Stochastic and Genetic Neural Network Combinations in Trading and Hybrid Time-Varying Leverage Effects

Georgios Sermpinis[a]
Charalampos Stasinakis[b]
Christian Dunis[c]

Abstract

The motivation of this paper is threefold. Firstly, we apply a Multi-Layer Perceptron (MLP), a Recurrent Neural Network (RNN) and a Psi-Sigma Network (PSN) architecture in a forecasting and trading exercise on the EUR/USD, EUR/GBP and EUR/CHF exchange rates and explore the utility of Kalman Filter, Genetic Programming (GP) and Support Vector Regression (SVR) algorithms as forecasting combination techniques. Secondly, we introduce a hybrid leverage factor based on volatility forecasts and market shocks and study if its application improves the trading performance of our models. Thirdly we introduce a specialized loss function for Neural Networks (NNs) in financial applications. In terms of our results, the PSN from the individual forecasts and the SVR from our forecast combination techniques outperform their benchmarks in statistical accuracy and trading efficiency. We also note that our trading strategy is successful, as it increased the trading performance of most of our models, while our NNs loss function seems promising.

[a] Corresponding Author -University of Glasgow Business School, University of Glasgow, Gilbert Scott Building, Glasgow, G12 8QQ (E-mail: georgios.sermpinis@glasgow.ac.uk)

[b] University of Glasgow Business School, University of Glasgow, Gilbert Scott Building, Glasgow, G12 8QQ (E-mail: c.stasinakis.1@research.gla.ac.uk)

[c] Horus Partners Wealth Management Group SA, Genève, Suisse and Liverpool Business School, JMU, John Foster Building , Liverpool L3 5UZ (E-mail: christian.dunis@hpwmg.com)

Keywords

Forecast Combinations, Kalman Filter, Genetic Programming, Support Vector Regression, Trading Strategies, Leverage.

1. INTRODUCTION

The term of Neural Network (NN) originates from the biological neuron connections of human brain. Artificial NNs are computation models that embody data-adaptive learning and clustering abilities, deriving from parallel processing procedures (Kröse and Smagt, 1996). NNs are considered a relatively new technology in Finance, but with high potential and an increasing number of applications (Lam (2004) and Bahrammirzaee (2010)). However, their practical limitations and contradictory empirical evidence have led to scepticism on whether they can outperform existing traditional models. NNs are similar to any advanced statistical model. They are optimized in an in-sample period and applied for prediction in an out-of-sample period. The difference of NNs with statistical models is on their adaptive nature. NNs can take many different forms and have as inputs any potential explanatory variable. Therefore they are capable of exploring different forms of non-linearity and theoretically provide a superior performance than statistical-econometrical models. Non-linearity is not possible to be measured in statistical terms and therefore models such as NNs have the advantage in problems where the exact nature of the series under study is unknown. Sceptics point out the NNs' lack of formal theoretical background and see them as a black box (Vellido *et al.* (1999) and Paliwal and Kumar (2009)). However, financial series and especially exchange rates are dominated by factors (e.g. behavioural factors, politics…) that time-series analysis and statistics are unable to capture in a single model. Based on this, it can be argued that a time-series statistical model that will capture the pattern of exchange rates in the long-run is impossible. Statistical theory and mathematics will never be able to explain such a complex relationship.

The Adaptive Market Hypothesis (AMH) argues that financial markets follow an evolutionary process. Profitable trading strategies exist at any time but their strength and robustness is diminishing over time (Lo (2004)). Heuristics (such as NNs, the GP and SVRs) try to imitate biological functions and create mathematical relationships. According to AMH they seem a perfect fit for our study and should outperform the classic statistical/technical models that dominate the relevant literature.

In this paper we examine the performance of a Multi-Layer Perceptron (MLP), a Recurrent Neural Network (RNN) and a Psi-Sigma Network (PSN) architecture in forecasting and trading the Euro/Dollar (EUR/USD), Euro/ British Pound (EUR/GBP) and Euro/Swiss Franc (EUR/CHF) exchange rate. Then, we explore the utility of Kalman Filter, Genetic Programming (GP) and Support Vector Regression (SVR) algorithms as forecasting combination techniques. As benchmarks for our NNs we use a Random Walk model (RW), an Autoregressive Moving Average model (ARMA) and a Smooth Transition Autoregressive Model (STAR). Our forecast combination techniques are then benchmarked by a Simple Average and a Least Absolute Shrinkage and Selection Operator (LASSO). Our forecasts are evaluated in terms of statistical accuracy and trading efficiency. All three exchange rates are highly liquid and well known for their high volatility in our days. Therefore, they are perfect series for a forecasting exercise with non-linear models.

The rationale of the paper is multiple. We explore if non-linear models such as NNs are able to outperform traditional models such as RW, ARMA and STAR. The STAR model will act as statistical non-linear benchmark while the comparison of our results with a RW model will add to the ongoing debate if financial forecasting models can outperform a RW[1]. In this forecasting competition we do not include structural macroeconomic models as presented by Flannery and Protopapadakis (2002), Andersen *et al.* (2003), Pearce and Solakoglu (2007), Evans and Speight (2010) and recently Bacchetta and Wincoop (2013). The main reason for that choice is the unavailability of daily data of relevant macroeconomic indicators. Comparing our models with benchmarks generated by lower frequency data would make the forecasting competition unfair and unequal. This study will also check if statistical models like the LASSO and the Kalman Filter can combine our forecasts successfully and provide a superior trading performance. Their results will be benchmarked against those generated by two advanced non linear techniques, a SVR and a GP model. SVR and GP algorithms have provided promising results in many fields of Science but they are rarely used as forecast combination techniques. The success of our best forecast combination model will be validated through the Modified Diebold-Mariano test (Harvey *et al.*, 1997). The trading strategy based on volatility forecasts will test if volatility forecasts and market shocks can be combined with our daily return forecasts to improve the trading performance of our models. The proposed loss function for NN models will add to the literature on the utility of NNs in Finance. Until now researchers are applying statistical loss functions to generate trading signals through NNs. However, statistical

[1] We also explored several autoregressive, moving averages and exponential smoothing models. Their statistical and trading performance was worse than those of ARMA for all exchange rates and periods considered.

accuracy is not always synonymous with financial profitability. Therefore, this function brings balance between these two terms.

All the forecasting models considered have unique characteristics, disadvantage and advantages. They will capture different aspects of the underlying pattern of the three exchange rates. By combining their forecasts we aim to generate signals that exploit the best elements of each single forecasting model and present a superior accuracy. Additionally the combined forecasts should be free from the biases of the individual models (as biases from opposite direction will counteract) and thereby lead to improved forecasting accuracy.

The rest of the paper is organized as follows. Section 2 is a literature review of previous relevant research on NNs and forecasting combination techniques. In Section 3 we give a detailed description of the three exchange rate series, used as our dataset, while Section 4 gives an overview of the benchmark models and the architectures of the NNs selected. Section 5 describes the forecast combination methods we implemented. The statistical and trading performance of our models is presented in Sections 6 and 7. Finally, some concluding remarks are summarized in Section 8.

2. LITERATURE REVIEW

Artificial Neural Networks (NNs) are computational models that embody data-adaptive learning and clustering abilities, deriving from parallel processing procedures (Kröse and Smagt, 1996). NNs seem to provide enough learning capacity and are more likely to capture the complex non-linear relationships which are dominant in the financial markets. Those advantages are well documented in the literature and a review of relevant studies is presented in De Gooijer's and Hyndman's (2006). However, skeptics argue that NNs present practical inefficiencies related to their 'parameter' tuning process and the generalization of their performance. For that reason, researchers apply either novel NN algorithms that try to overcome some of these limitations (Ling *et al.*(2003) or forecast combination techniques that seem able to combine the virtues of different networks for superior forecasts (see amongst others Harrald and Kamstra (1997), and Teräsvirta *et al.* (2005)).

Many researchers have attempted to forecast exchange rates, but their empirical results are often contradictory. Meese and Rogoff (1983a, 1983b) examine the Frenkel-Bilson, Dornbusch-Frankel, and Hooper-Morton structural exchange rate models and find that the random walk performs better. The authors conclude that the out-of-sample failure of these models is due to the volatile nature of exchange rates, the poor inflation measurements and their money demand misspecifications. On the other hand, Tenti (1996) presents promising results in predicting the exchange rate of the Deutsche Mark with three different RNN architectures. Bissoondeeal *et al.* (2008) use linear and nonlinear methods in forecasting AUD/USD and GBP/USD exchange rates and conclude that NNs outperform the traditional ARMA and GARCH models. Finally, Grossmann and McMillan (2010) propose a time-varying ESTR equilibrium exchange rate model for forecasting the bilateral rates between the US Dollar and the Canadian Dollar, the Japanese Yen and the British Pound. Their non-linear model provides superior forecasts in terms of directional change accuracy when compared to their linear alternatives.

The above studies forecast exchange rates, but they do not use their predictions for trading purposes. Park and Irwin (2007) report that the number of successful trading studies of foreign exchange rates increase rapidly after 2000, although there are some earlier signs of marginal FX profitability (Lee and Mathur, 1996). Neely (2002) examines the DEM/USD, DEM/USD, JPY/USD, CHF/USD and AUD/USD exchange rates through a moving average and the means of the estimated returns do not exceed 9%. Fernández-Rodríguez *et al.* (2003) suggest that trading signals deriving from a nearest neighbor algorithm are superior to moving average rules in European exchange markets during 1978–1994. Their non-linear trading rule on Danish krona, French franc, Dutch guilder and Italian lira, generate returns from 1.5% to 20.1%. Chen and Leung (2004) with linear and non linear models achieve profits less than 10% for three different exchange rates. Qi and Wu (2006) examine seven foreign exchange rates during 1973–1998 and achieved with simple technical trading rules generate profits from 7.2% to 12.2% after transaction costs. Dueker and Neely (2007) apply a Markov switching trading rule on four exchange rates and achieve out-of-sample annualized returns from 1.02% to 7.54%. Dunis *et. al.* (2010) and Sermpinis *et. al.* (2012) conduct trading exercises with NNs on the EUR/USD exchange rate. Their models achieve an annualized return of 5% to 17%.

In this context, profitability in the FX markets seems possible, but the profits are volatile given the periods under study and the models implemented. This study attempts to overcome this instability by combining the successful trading signals of NN models with stochastic and genetic algorithms. The results of this paper show that the forecast combinations achieve profits more than 20% and remain consistent for three exchange rates and three rolling exercises. This performance further improves with a hybrid leverage trading strategy.

Forecast combination approaches that aim to achieve higher levels of profits are limited in the literature. Nonetheless, the idea of combining forecasts to improve prediction accuracy in not new. It originates from Bates and Granger (1969), who suggested combining rules based on variances-covariances of the individual forecasts. Since then, many forecasting combination methods have been proposed and applied in financial research. Donaldson and Kamstra (1999) use combination techniques, such as weighted OLS, to benchmark the performance of artificial NN forecasts of S&P 500 stock index and conclude that the NNs are more statistically accurate. Hu and Tsoukalas (1999) combine the individual volatility forecasts of four models with simple averaging, ordinary least squares model and a NN. Their result suggest that the NN combination model performed better during the August 1993 crisis, especially in terms of root mean absolute forecast error. De Menezes and Nikolaev (2006) present promising forecasting results with their polynomial neural network forecasting system, which combines genetic programming with NN models. Altavilla and De Grauwe (2008) compare the performance of linear and nonlinear models in forecasting exchange rates. Although linear models are better at short forecasting horizons and nonlinear models dominate at longer forecasting horizons, they suggest that combining different forecasting techniques generally produces more accurate forecasts. Guidolin and Timmermann (2009) combine forecasts of future spot rates with forecasts of macroeconomic variables and conclude that this improves the out-of-sample forecasting performance of US short-term rates. Andrawis *et al.* (2011) attempt to predict the daily cash withdrawal amounts from ATM machines. In their application, they forecast over one hundred time series with eight classes of linear and non-linear models. Their results show that a simple average of NN, Gaussian process regression and linear models' forecasts is the optimal. Ebrahimpour *et al.* (2011) apply and compare three NN combining methods and an Adaptive Network-Based Fuzzy Inference System to trend forecasting in the Tehran stock exchange. The mixture of MLP experts is the model that presents the best hybrid model in this competition, but all NN combining models present promising forecasting performance.

3. THE EXCHANGE RATES AND RELATED FINANCIAL DATA

The European Central Bank (ECB) publishes a daily fixing for selected EUR exchange rates. These reference mid-rates are based on a daily concentration procedure between central banks within and outside the European System of Central Banks, which normally takes place at 2.15 p.m. ECT time. The reference exchange rates are published both by electronic market information providers and on the ECB's website shortly after the concentration procedure has been completed. Although only a reference rate, many financial institutions are ready to trade at the EUR fixing and it is therefore possible to leave orders with a bank for business to be transacted at this level. The ECB daily fixings of the EUR exchange rate are therefore tradable levels and using them is a more realistic alternative to, say, London closing prices.

In this paper, we examine the EUR/USD, EUR/GBP and EUR/CHF over the period of 1999-2012 in three rolling forecasting exercises (F1, F2 and F3) on a daily basis. Each exercise studies a decade of the respective exchange rate using the last two years for out-of-sample evaluation. F1 focus on the decade of 1999-2008 while F2 and F3 examine the periods 2001-2010 and 2003-2012 respectively. Table 1 below presents the three sub-periods.

[Insert Table 1]

The in-sample datasets for each exercise are further divided in two sub-periods, the training and test sub-period. This is done for training purposes of our NNs. More details on this issue can be found in section 4.2.

The three rolling forward sub-periods add validity to the forecasting exercise and increase the robustness of our results. The out-of-sample periods are dominated by the effects of the debt and the mortgage crises. By rolling forward the estimation we attempt to capture the effect of these crises to the extent that is possible. The rolling forward estimation and the fact that the parameterization of our model is conducted entirely in-sample will also act as a shield against the data-snooping bias.

The graph below shows the total datasets of the EUR/USD, EUR/GBP and EUR/CHF and their volatile trend. The out-of-sample periods of each exercise are also highlighted.

[Insert Figure 1]

The time series, shown above, are non-normal and non-stationary. To overcome the non-stationary issue, every series is transformed into a daily series of rate returns. So given the price level P_1, P_2, ..., Pt the return[2] at time t is calculated as:

$$R_t = \left(\frac{P_t}{P_{t-1}}\right) - 1 \qquad (1)$$

The Jarque-Bera statistic and the ADF values confirm that the return series are non-normal and non-stationary at the 99% confidence interval. The descriptive statistics of the three exchange rate series and the ADF p-values are presented in Appendix A.

4. FORECASTING MODELS

4.1 BENCHMARK FORECASTING MODELS

In this paper, we use three traditional forecasting strategies, namely a Random Walk (RW), an Autoregressive Moving Average model (ARMA) and a Smooth Transition Autoregressive Model (STAR) in order to benchmark the efficiency of the NN models. The aim of our models is to forecast the one day ahead return of the series under study for each forecasting exercise.

4.1.1 Random Walk Model (RW)

The RW is a process where the current value of a variable is calculated from the past value plus an error term. The error term follows the standard normal distribution. The specification of the model is:

$$\hat{Y}_t = Y_{t-1} + e_t, \; e_t \sim N(0,1) \qquad (2)$$

Where $\hat{Y}_t$ is the forecasted value for period t and Y_{t-1} is the actual value of period *t-1*.

[2] For small returns as we have in our application, the arithmetic and the logarithmic returns are almost identical (Rozeff and Kinney (1976)). Also log-returns are not linear additive across portfolio components, which can be a problem. Market participants tend to look at discrete returns in their daily trading activity, thus making the use of arithmetic returns more realistic.

The RW is a non-stationary process with a constant mean, but not a constant variance.

4.1.2 Auto-Regressive Moving Average Model (ARMA)

The ARMA model is based on the assumption that the current value of a time-series is a linear combination of its previous values plus a combination of current and previous values of the error terms (Brooks, 2008). Thus, the ARMA model embodies autoregressive and moving average components and can be specified as below:

$$\hat{Y}_t = \varphi_0 + \varphi_1 Y_{t-1} + \varphi_2 Y_{t-2} + \ldots + \varphi_p Y_{t-p} + \varepsilon_t - w_1 \varepsilon_{t-1} - w_2 \varepsilon_{t-2} - \ldots - w_q \varepsilon_{t-q} \quad (3)$$

Where:

- $\hat{Y}_t$ is the forecasted value at time t
- $Y_{t-1}, Y_{t-2}, \ldots Y_{t-p}$ are the lagged actual values
- $\varphi_0, \varphi_1, \ldots, \varphi_p$ are the regression coefficients
- ε_t is the error term
- $\varepsilon_{t-1}, \varepsilon_{t-2}, \ldots, \varepsilon_{t-q}$ are the previous values of the error terms
- $w_1, w_2, \ldots, w_q$ are the error weights

Using as a guide the information criteria in the in-sample subset the optimal ARMA structures were selected. In all cases, the null hypotheses that all coefficients (except the constant) are not significantly different from zero and that the error terms are normally distributed are rejected at the 95% confidence interval. The orders of the restricted ARMA structures used for each exchange rate and forecasting exercise are presented in following table.

[Insert Table 2]

4.1.3 Smooth Transition Autoregressive Model (STAR)

STARs initially proposed by Chan and Tong (1986) are extensions of the traditional autoregressive models (ARs). The STAR combines two AR models with a function that defines the degree of non-linearity (smooth transition function). The general two-regime STAR specification is the following:

$$\hat{Y}_t = \Phi_1' \mathrm{X}_t (1 - F(z_t, \zeta, \lambda)) + \Phi_2' \mathrm{X}_t F(z_t, \zeta, \lambda) + u_t \quad (4)$$

Where:

- $\hat{Y}_t$ the forecasted value at time t
- $\Phi_i = (\tilde{\varphi}_{i,0}, \tilde{\varphi}_{i,1}, ... \tilde{\varphi}_{i,p}), i = 1,2$ and $\tilde{\varphi}_{i,0}, \tilde{\varphi}_{i,1}, ... \tilde{\varphi}_{i,p}$ the regression coefficients of the two AR models
- $X_t = (1, \tilde{\chi}_t')'$ with $\tilde{\chi}_t' = (Y_{t-1}, ..., Y_{t-p})$
- $0 \leq F(z_t, \zeta, \lambda) \leq 1$ the smooth transition function
- $z_t = Y_{t-d}, d > 0$ the lagged endogenous transition variable
- ζ the parameter that defines the smoothness of the transition between the two regimes
- λ the threshold parameter
- u_t the error term

In this paper we follow the steps of Lin and Teräsvirta (1994) in order to determine when the series is best modeled as a Logistic STAR or an Exponentional STAR process.

4.2 NEURAL NETWORKS (NNs)

The use of NNs in financial forecasting is not new and several researchers have successfully applied them to the task of identifying patterns in price time series or estimating the profitability of technical trading rules. In this study three NNs architectures (the MLP, the RNN and the PSN) are applied to the task of forecasting and trading three exchange rates. All these architectures have at least three layers. The first layer is called the input layer (the number of its nodes corresponds to the number of explanatory variables). The last layer is called the output layer (the number of its nodes corresponds to the number of response variables). An intermediary layer of nodes, the hidden layer, separates the input from the output layer. Its number of nodes defines the amount of complexity the model is capable of fitting. In addition, the input and hidden layer contain an extra node called the bias node. This node has a fixed value of one and has the same function as the intercept in traditional regression models. Normally, each node of one layer has connections to all the other nodes of the next layer. The training of the network (which is the adjustment of its weights in the way that the network maps the input value of the training data to the corresponding output value) starts with randomly chosen weights and proceeds by applying a learning algorithm called backpropagation of errors (Shapiro, 2000). The iteration length is optimised by maximising a fitness function in the test dataset.

Unlike MLPs, RNNs have an activation feedback which embodies short-term memory. In other words, the RNN architecture can provide more accurate outputs because the inputs are (potentially) taken from all previous values. Tenti (1996) notes that RNNs need more connections and memory than standard backpropagation networks. However, RNNs can yield better results in comparison with simple MLPs due to the additional memory inputs. The PSN model was firstly introduced by Shin and Ghosh (1991). They are a class of feed-forward fully connected higher order NNs, which require less number of weights and processing units for their training. Their main advantage is that they combine the fast learning property of single layer networks with the powerful mapping capability of higher order NNs, while avoiding the combinatorial increase in the required number of weights. The order of the network in the context of PSNs is represented by the number of hidden nodes. In a PSN the weights from the hidden to the output layer are fixed to one and only the weights from the input to the hidden layer are adjusted, something that greatly reduces the training time. The activation function of the nodes in the hidden layer is the summing function, while the activation function of the output layer is a sigmoid one. For more information on MLP, RNN and PSN architectures see Zhang *et al.* (1998), Ghazali *et al.* (2006) and Sermpinis *et al.* (2012).

For training purposes of our NNs, we further divide our in-sample dataset in two sub-periods, the training and test sub-period. In the absence of any formal theory behind the input selection of NNs, we conduct some NN experiments and a sensitivity analysis on a pool of potential inputs in the in-sample dataset in order to help our decision. In our application, we experimented in the training sub-period and we selected as inputs the set of variables that provided the higher trading performance in the test sub-period. This optimization procedure is the most popular in NNs and superior to cross validation for datasets of our size (Zhu and Rohwer (1996)). The sets of inputs for the F1, F2 and F3 of each exchange rate are presented in Table 3 below[3].

[Insert Table 3]

For our NNs which are specially designed for financial purposes, we apply a novel two-objective fitness function. This fitness function focuses on achieving two goals at the same time. First of all, the annualized return in the test period should be maximized and secondly the Root Mean Square Error (RMSE) of the

[3] We also explored as inputs autoregressive terms of other exchange rates (e.g. the USD/JPY and GBP/JPY exchange rates), commodities prices (e.g. Gold Bullion and Brent Oil) and stock market prices (e.g. FTSE100 and DJIA). However, the set of inputs presented in Table 3 gave our NNs the highest trading performance in the in-sample period during our sensitivity analysis and were thus retained. Macroeconomic variables are not used as inputs in this study due to their monthly and quarterly frequency.

networks output should be minimized. Based on the above the fitness function for all our NNs takes the form below[4] and equation (5) is maximized:

$$\text{Fitness} = \text{Annualized\_Return} - 10*\text{RMSE} \tag{5}$$

After our networks are optimized, the predictive value of each model is evaluated by applying it to the validation dataset (out-of-sample dataset). Since the starting point for each network is a set of random weights, forecasts can slightly differ between same networks. In order to eliminate any variance between our NN forecasts and add robustness to our results, we used the simple average of a committee of 10 NNs which presented the highest profit in the in-sample sub-period. This was necessary in order to eliminate any outlier network that could jeopardise our conclusions. The characteristics of the NNs used in this paper for each forecasting exercise and exchange rate are given in Appendix B.

Several NNs trading applications suffer for the data snooping effect. Data-snooping occurs when a given set of data is used more than once. This can leads to the possibility that the results achieved may be due to chance rather than an inherent merit in the method. In order to avoid this effect we follow the guidelines of James *et al.* (2012) and as stated above, we clearly subdivide our data in in-sample (training and test subsets) and out-of-sample (validation subset). The out-of-sample subset is not used in any part of our NN parameter selection procedure. In order to validate that our NN out-of-sample forecasts are free from the data snooping bias, we apply the Hansen (2005) test. As benchmark for this comparison, we use a simple martingale model. Our results indicate that the NNs committees (which forecasting performance is presented in the following sections) are free from the data snooping bias at the 5% level in all out-of-sample subsets.

5. FORECASTING COMBINATION TECHNIQUES

In this section we present the techniques that we used to combine our NNs forecasts. It is important to outline that a forecast combination targets either to follow the trend of the best individual forecast ('combining for adaptation') or to significantly outperform each one of them ('combining for improvement') (Yang, 2004). Consequently, we decided to exclude the RW and the ARMA strategies from

[4] The RMSE is multiplied by 10 so the two factors in our equation are more or less equal in levels.

our combination techniques. Those strategies present a considerably worse trading performance than their NNs' counterparts both in-sample and out-of-sample. Therefore, their inclusion in our combination techniques will deteriorate their performance, rather than improve it.

5.1 SIMPLE AVERAGE

The first forecasting combination technique used in this paper is the Simple Average, which can be considered a benchmark forecast combination model. Given the three NNs' forecasts $f_{MLP}^{t}, f_{RNN}^{t}, f_{PSN}^{t}$ at time t, the combination forecast at time t is calculated as: $f_{c_{NNs}}^{t} = (f_{MLP}^{t} + f_{RNN}^{t} + f_{PSN}^{t}) / 3$ (6)

5.2 LEAST ABSOLUTE SHRINKAGE AND SELECTION OPERATOR (LASSO)

The LASSO Regression is a class of Shrinkage or Regularization Regressions, which applies when multicollinearity exists among the regressors (Sundberg, 2006). The main difference between this technique and the Ordinary Least Squares (OLS) Regression is that the LASSO method also minimizes the residual squared error, by adding a coefficient constraint (similarly to a Ridge Regression (Chan *et al.*, 1999)). Compared to Ridge Regression, LASSO best applies in samples of few variables with medium/large effect, as in our case (Hastie *et al.*, 2009). For more details on the mathematical specifications of LASSO see Wang *et al.* (2007).

Given the vectors of independent and dependent variables:

$$\begin{pmatrix} X_1^T \\ \vdots \\ X_N^T \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NN} \end{pmatrix}, \quad Y = (y_1, \ldots, y_N)^T \tag{7}$$

and the training data *{(X_1,y_1),…,(X_N,y_N)}*, the LASSO coefficients are estimated based on the following argument:

$$\hat{\beta}_{lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_i x_{ij} \right)^2 \right\} \quad \textit{subject to} \quad \sum_{j=1}^{d} \left| \beta_j \right| \le k, k > 0 \tag{8}$$

The argument (8) is based on Breiman's non-negative garrotte minimization process (Yuan and Lin, 2007). Here k stands for the "*tuning parameter"*, since it controls the amount of shrinkage applied to the coefficients (Tibshirani, 2011). In our case, we experimented with various values of k in the in-sample period and we concluded that the best results in terms of trading performance are acquired when the constraints take the following forms for EUR/USD, EUR/GBP and EUR/CHF respectively:

$$\begin{Bmatrix} F1_{USD} : |\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| \leq 3.1 \\ F2_{USD} : |\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| \leq 1.9 \\ F3_{USD} : |\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| \leq 2.3 \end{Bmatrix} \tag{9}$$

$$\begin{Bmatrix} F1_{GBP} : |\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| \leq 2.5 \\ F2_{GBP} : |\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| \leq 3.8 \\ F3_{GBP} : |\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| \leq 3.1 \end{Bmatrix} \tag{10}$$

$$\begin{Bmatrix} F1_{CHF} : |\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| \leq 1.9 \\ F2_{CHF} : |\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| \leq 2.2 \\ F3_{CHF} : |\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| \leq 4.7 \end{Bmatrix} \tag{11}$$

Subject to the above, the final LASSO forecast combinations are given by the following set of equations for EUR/USD, EUR/GBP and EUR/CHF respectively:

$$\begin{Bmatrix} F1_{USD} : f^t_{c_{NNs}} = 0.125 f^t_{MLP} + 0.723 f^t_{RNN} + 1.534 f^t_{PSN} + \varepsilon_t \\ F2_{USD} : f^t_{c_{NNs}} = 0.021 f^t_{MLP} + 0.218 f^t_{RNN} + 1.452 f^t_{PSN} + \varepsilon_t \\ F3_{USD} : f^t_{c_{NNs}} = 0.095 f^t_{MLP} + 0.314 f^t_{RNN} + 1.684 f^t_{PSN} + \varepsilon_t \end{Bmatrix} \tag{12}$$

$$\begin{Bmatrix} F1_{GBP} : f^t_{c_{NNs}} = 0.251 f^t_{MLP} + 0.871 f^t_{RNN} + 1.851 f^t_{PSN} + \varepsilon_t \\ F2_{GBP} : f^t_{c_{NNs}} = 0.105 f^t_{MLP} + 0.355 f^t_{RNN} + 1.681 f^t_{PSN} + \varepsilon_t \\ F3_{GBP} : f^t_{c_{NNs}} = 0.201 f^t_{MLP} + 0.488 f^t_{RNN} + 1.745 f^t_{PSN} + \varepsilon_t \end{Bmatrix} \tag{13}$$

$$\begin{Bmatrix} F1_{CHF} : f^t_{c_{NNs}} = 0.161 f^t_{MLP} + 0.735 f^t_{RNN} + 1.486 f^t_{PSN} + \varepsilon_t \\ F2_{CHF} : f^t_{c_{NNs}} = 0.269 f^t_{MLP} + 0.412 f^t_{RNN} + 1.318 f^t_{PSN} + \varepsilon_t \\ F3_{CHF} : f^t_{c_{NNs}} = 0.157 f^t_{MLP} + 0.856 f^t_{RNN} + 1.952 f^t_{PSN} + \varepsilon_t \end{Bmatrix} \tag{14}$$

Each constraint makes the model adaptive, since it creates a penalization balance on each estimate by leading some coefficients to zero or close to zero.

5.3 KALMAN FILTER

Kalman Filter is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements. The time-varying coefficient combination forecast suggested in this paper is shown below:

$$\text{Measurement Equation:} \quad f_{c_{NNs}}^{t} = \sum_{i=1}^{3} a_i^t f_i^t + \varepsilon_t, \quad \varepsilon_t \sim NID\left(0, \sigma_\varepsilon^2\right) \tag{15}$$

$$\text{State Equation:} \quad a_i^t = a_i^{t-1} + n_t, \quad n_t \sim NID(0, \sigma_n^2) \tag{16}$$

Where:

- $f_{c_{NNs}}^{t}$ is the dependent variable (combination forecast) at time t
- f_i^t $(i = 1, 2, 3)$ are the independent variables (individual forecasts) at time t
- a_i^t $(i = 1, 2, 3)$ are the time-varying coefficients at time t for each NN
- ε_t, n_t are the uncorrelated error terms (noise)

When Kalman Filter is applied, all a_i^t are estimated in time, along with the log-likelihood of the model based on the observations up to time t. Then the likelihood function is maximized with a numerical optimization algorithm, based on σ_n^2. The updated alphas for the state equation are estimated at time t based on the new observations at time t and then the state estimates are propagated in time $t+1$. Thus, the Kalman Filter update can be considered as the best unbiased linear estimate of the individual forecasts f_i^t, given $f_{c_{NNs}}^{t}$ and the prior information. After the Kalman Filter and the numerical optimization algorithm, a Kalman smoothing algorithm should be applied, because the accuracy is increased to the end of the sample. This algorithm 'smoothes' the estimates by running backwards in time and using information acquired after time t and allows our model to compute forecasts which use all available measurement data over the forecast sample. Following Welch and Bishop (2001), in our study the alphas are calculated by a simple random walk and we initialized $\varepsilon_1 = 0$. Based on the above, our Kalman Filter model provides the following final states for the respective exercises and exchange rates:

$$\left\{ \begin{array}{l} F1_{USD}: f_{c_{NNs}}^{t} = 0.152 f_{MLP}^{t} + 0.785 f_{RNN}^{t} + 1.485 f_{PSN}^{t} + \varepsilon_t \\ F2_{USD}: f_{c_{NNs}}^{t} = 0.081 f_{MLP}^{t} + 0.976 f_{RNN}^{t} + 1.322 f_{PSN}^{t} + \varepsilon_t \\ F3_{USD}: f_{c_{NNs}}^{t} = 0.108 f_{MLP}^{t} + 0.655 f_{RNN}^{t} + 1.271 f_{PSN}^{t} + \varepsilon_t \end{array} \right\} \tag{17}$$

$$\begin{cases} F1_{GBP} : f^{t}_{c_{NNs}} = 0.134 f^{t}_{MLP} + 0.622 f^{t}_{RNN} + 1.592 f^{t}_{PSN} + \varepsilon_t \\ F2_{GBP} : f^{t}_{c_{NNs}} = 0.102 f^{t}_{MLP} + 0.813 f^{t}_{RNN} + 1.628 f^{t}_{PSN} + \varepsilon_t \\ F3_{GBP} : f^{t}_{c_{NNs}} = 0.196 f^{t}_{MLP} + 0.738 f^{t}_{RNN} + 1.345 f^{t}_{PSN} + \varepsilon_t \end{cases} \quad (18)$$

$$\begin{cases} F1_{CHF} : f^{t}_{c_{NNs}} = 0.121 f^{t}_{MLP} + 0.572 f^{t}_{RNN} + 1.701 f^{t}_{PSN} + \varepsilon_t \\ F2_{CHF} : f^{t}_{c_{NNs}} = 0.131 f^{t}_{MLP} + 0.663 f^{t}_{RNN} + 1.586 f^{t}_{PSN} + \varepsilon_t \\ F3_{CHF} : f^{t}_{c_{NNs}} = 0.182 f^{t}_{MLP} + 0.595 f^{t}_{RNN} + 1.263 f^{t}_{PSN} + \varepsilon_t \end{cases} \quad (19)$$

From the above, we note that the Kalman filtering process, as in the case of LASSO, also favors PSN forecasts regardless the period under study. This is what one would expect, since it is the model that performs best individually. In order to achieve optimal Kalman Filter estimation, it is important though to introduce a noise ratio $n_r = \sigma_\varepsilon^2 / \sigma_n^2$ (20). The results are becoming more adaptive when the noise ratio increases. When $\sigma_n^2 = 0$, the model transforms to the typical OLS model.

5.4 GENETIC PROGRAMMING ALGORITHM (GP)

GP algorithms are a class of Genetic Algorithms (GAs) and the intuition behind this technique is the Darwinian principle of reproduction and survival of the fittest. Thus GP applies the Darwinian theory of evolution to a population of computer programs of varying sizes and shapes, which run in various environments in order to produce forecasts at a high level of accuracy (Chen, 2002). GP creates a random initial population of models and evolves it using genetic operators, in order to calculate the mathematical expression which best fits the specified data input in the system.

Our GP application evolves tree-based structures that present models (sub-trees) of input – output. It utilizes formulas to evolve algebraic expressions that enable the analysis and optimization of results in a genetic tree structure. This structure consists of nodes, which are essentially functions that perform actions within this structure. The maximum tree depth is the maximum length of each model (of each tree structure) and it depends on the functions and terminals of each individual model. The NNs' individual forecasts are used as inputs, while the nodes' functions are in place to generate output signals. In order to limit the 'bloat effect', a similar issue as overfitting in NNs, we follow the parsimony pressure method (Koza (1992)). The selection probabilities of an individual are dependent on the size of the program. The bigger the individual, the lower is his selection probability and the possibility to have children and survive through the generations. GP reproduces newer models replacing the weaker ones in the population according to their fitness. In our case the fitness function is:

$$\text{MSE-c(t)*size}(x_t) \tag{20}$$

Where MSE is Mean Squared Error of the forecasted return and the actual return of each day, c(t) is a variable that is dependent on the size of all individuals in generation t and size(x_t) is the size of individual x in generation t. For more detail on the parsimony pressure method and the variable c(t) see Koza (1992) and Poli and McPhee (2008).

Our algorithm is conducting tournaments and selects the best models of each generation. These models are exposed to two genetic operators, known as mutation and crossover. Mutation is the creation of a new model that is mutated randomly from an existing one. This is calibrated in our model by setting a mutation probability. Crossover is the creation of two new models from existing ones by genetically recombining randomly chosen parts of them. These genetic procedures produce superior offsprings that will replace the worst models (tournament losers) and rearrange the initial population for the next iteration.

These steps are repeated until the predefined termination criterion for genetic programming is satisfied. In this paper the termination criterion is set to 100,000 at which point the cycles are stopped and forecasted results can be obtained. For more details on the functionality aspects of GP and the genetic operators see Koza (1992). The figure below describes the structure of a typical GP algorithm.

[Insert Figure 2]

The parameters of our GP application are presented in Appendix C.

5.5 SUPPORT VECTOR REGRESSION (SVR)

Support Vector Machine (SVM) is a well-known approach in the machine learning community. It was originally developed for solving classification problems in supervised learning frameworks. The introduction of the ε-sensitive loss function by Vapnik (1995) though established Support Vector Regression (SVR) as a robust technique for constructing data-driven and non-linear empirical regression models. Recently SVR and its hybrid applications have become popular for time-series prediction and financial forecasting applications. They provide global and unique solutions and do not suffer from multiple local minima (Suykens, 2002) while SVRs seem also able to cope well with high-dimensional, noisy and complex feature problems. Moreover, they present a remarkable ability of balancing model accuracy and model complexity, depending on the available data (see amongst others Montana and Parrella (2008) and Lu *et al.*(2009)).

5.5.1 The ε-SVR

Considering the training data $\{(x_1,y_1), (x_2,y_2)..., (x_n, y_n)\}$, where $x_i \in X \subseteq R, y_i \in Y \subseteq R, i = 1...n$ and n the total number of training samples, the SVR function can be specified as: $f(x) = w^T \varphi(x) + b$ (21)

where w and b are the regression parameter vectors of the function and $\varphi(x)$ is the non-linear function that maps the input data vector x into a feature space where the training data exhibit linearity (see Figure 3c). The ε-sensitive loss L_ε function finds the predicted points that lie within the tube created by two slack variables ξ_i, ξ_i^*

$$L_\varepsilon(x_i) = \begin{cases} 0 \; if \; | y_i - f(x_i) | \leq \varepsilon \\ | y_i - f(x_i) | - \varepsilon \; if \; other \end{cases}, \varepsilon \geq 0 \tag{22}$$

In other words ε is the degree of model noise insensitivity and L_ε finds the predicted values that have at most ε deviations from the actual obtained values y_i and (see Figure 3a and 3b).

[Insert Figure 3]

The goal is to solve the following argument:

$$\text{Minimize } \; C\sum_{i=1}^{n}(\xi_i + \xi_i^*) + \frac{1}{2}\|w\|^2 \text{ subject to } \begin{Bmatrix} \xi_i \geq 0 \\ \xi_i^* \geq 0 \\ C > 0 \end{Bmatrix} \text{and } \begin{Bmatrix} y_i - w^T\varphi(x_i) - b \leq +\varepsilon + \xi_i \\ w^T\varphi(x_i) + b - y_i \leq +\varepsilon + \xi_i^* \end{Bmatrix} \tag{23}$$

The above quadratic optimization problem is transformed in a dual problem and its solution is based on the introduction of two Lagrange multipliers a_i, a_i^* and mapping with a kernel function $K(x_i, x)$:

$$f(x) = \sum_{i=1}^{n}(a_i - a_i^*)K(x_i, x) + b \text{ where } 0 \leq a_i, a_i^* \leq C \tag{24}$$

Factor b is computed following the Karush-Kuhn-Tucker conditions (for a detailed mathematical analysis of the above solution see Vapnik (1995)). Support Vectors (SVs) are called all the x_i that contribute to equation (24), thus they lie outside the ε-tube, whereas non-SVs lie within the ε-tube.[5] Increasing ε leads to less SVs' selection, whereas decreasing it results to more 'flat' estimates. The norm term $\|w\|^2$ characterizes the complexity (flatness) of the model and the term $\left\{\sum_{i=1}^{n}(\xi_i+\xi_i^*)\right\}$ is the training error, as specified by the slack variables. Consequently the introduction of the parameter C satisfies the need to trade model complexity for training error and vice versa (Cherkassky and Ma, 2004).

5.5.2 The *v*-SVR

The v-SVR algorithm encompasses the ε parameter in the optimization process and controls it with a new parameter $v \in (0,1)$ (Basak *et al.*, 2007). In v-SVR the optimization problem transforms to:

$$\text{Minimize } C\left(v\varepsilon+\frac{1}{n}\sum_{i=1}^{n}(\xi_i+\xi_i^*)\right)+\frac{1}{2}\|w\|^2 \text{ subject to } \begin{Bmatrix}\xi_i \geq 0\\ \xi_i^* \geq 0\\ C>0\end{Bmatrix} \text{ and } \begin{Bmatrix} y_i - w^T\varphi(x_i)-b \leq +\varepsilon+\xi_i \\ w^T\varphi(x_i)+b-y_i \leq +\varepsilon+\xi_i^*\end{Bmatrix} \tag{25}$$

The methodology remains the same as in ε-SVR and the solution takes a similar form:

$$f(x)=\sum_{i=1}^{n}(a_i-a_i^*)K(x_i,x)+b \text{ where } 0 \leq a_i, a_i^* \leq \frac{C}{n} \tag{26}$$

Based on the 'v-trick', as presented by Schölkopf *et al.* (1999), increasing ε leads to the proportional increase of the first term of $\left\{v\varepsilon+\frac{1}{n}\sum_{i=1}^{n}(\xi_i+\xi_i^*)\right\}$, while its second term decreases proportionally to the fraction of points outside the ε-tube. So v can be considered as the upper bound on the fraction of errors. On the other hand, decreasing ε leads again to a proportional change of the first term, but also the second term's change is proportional to the fraction of SVs. That means that ε will shrink as long as the fraction of SVs is smaller than v, therefore v is also the lower band in the fraction of SVs.

5.5.3 SVR Parameter Selection

[5] A SV is either a boundary vector ($(a_i-a_i^*) \in [-C/n, C/n], \xi_i=\xi_i^*=0$) or an error vector ($a_i, a_i^* = C/n \text{ and } \xi_i, \xi_i^* > 0$).

The lack of information on the noise of the training datasets makes the *a priori* ε-margin setting a difficult task. In general, there are no optimal solutions to this problem, but four approaches can be identified as most common in the literature:

- Setting ε as a non-negative constant for convenience ($\varepsilon=0$ or equal to a very small value) (Trafalis and Ince (2000)).
- Choosing ε by maximizing the statistical efficiency of a location parameter estimator, as presented by Smola *et al.* (1998).
- Estimating ε with the cross-validation technique (Cao *et al.*(2003)).
- Controlling ε with v-SVR (Schölkopf *et al.* (1999) and Basak *et al.* (2007)).

In this paper we implement a RBF v-SVR, thus we have to determine two kernel-independent parameters (v and C) and the RBF parameter (γ) of the RBF Kernel function: $K(x_i, x) = \exp(-\gamma \|x_i - x\|^2), \gamma > 0$ (27)

RBF kernels are the most common in similar SVR applications (see and Ince and Trafalis (2006 and 2008)). This is based on the fact that they efficiently overcome overfitting and seem to excel in directional accuracy. Having made that choice of Kernel, we are able to follow Cherkassky's and Ma's (2004) RBF application of optimal choice of C through a standard parameterization of the SVR solution. Based on their approach:

$$|f(x)| \leq \left| \sum_{i=1}^{n_{sv}} (a_i - a_i^*) K(x_i, x) \right| \leq \sum_{i=1}^{n_{sv}} |(a_i - a_i^*)| \cdot |K(x_i, x)| \leq \sum_{i=1}^{n_{sv}} C \cdot |K(x_i, x)| \qquad (28)$$

For $K(x_i, x) = \exp(-\gamma \|x_i - x\|^2) \leq 1$, we obtain the upper bound of the SVR function as $|f(x)| \leq C \cdot n_{sv}$. Thus, the estimation of C independently of the number of support vectors n_{sv} is given by $C \geq |f(x)|$ for all training samples. In other words, the optimal choice of C is equal to the range of the output values of our training data. In order to overcome outliers, the final C is computed as $C = \max(|\bar{y} + 3\sigma_y|, |\bar{y} - 3\sigma_y|)$ (29) where $\bar{y}, \sigma_y$ is the mean and the standard deviation of the training responses respectively. Based on that we calculate the parameters for the scenario of EUR/USD as $C^{F1}=0.02$, $C^{F2}=0.022$ and $C^{F3}=0.002$.

In most SVR studies, the model parameters are determined one at a time by letting each parameter taking a range of different values and then identifying the value that corresponds to the best model performance assessed by cross-validation (see Chalimourda *et al.* (2004) and Smola and Schölkopf (2004)). In our case, we apply a 5-fold cross-validation for calculating the optimal v and γ in our in-sample dataset, having set the parameter C for the respective exercise. During this cross-validation process, we partition the in-sample period into five equal subsamples. From those subsamples, a single subsample is retained as the validation data for testing the model and the remaining four subsamples are used as training data. The cross-validation process is then repeated five times with each one of the subsamples used only once as the validation data. As suggested by Duan *et al.* (2003) keeping the number of folds moderate, i.e. five, offers efficient parameter estimation with constraining substantially computational costs.

For example, regarding the exercise *F1* of the EUR/USD the cross-validation is performed for the v parameter with $C^{F1}=0.02$ and fixed values of γ^{F1}. Our selection is based on the best trading performance in the in-sample dataset. Nonetheless, the value of the parameter γ^{F1} is not constrained. In order to overcome this issue, our model encompasses a *pseudo-R^2* criterion (Veall and Zimmermann, 1996). This criterion is calculated based on the residual sum of squared errors of each model (RSS_v) and a default model (RSS_{def}). "Default model" is the model which does not use information from the independent variables for the prediction of the dependent variable. According to the least square principle, the default model is simply the mean of the dependent variable computed in the training sample:

$$pseudo-R^2 = 1-\frac{RSS_v}{RSS_{def}} \text{ where } RSS_{def} = \sum (y_i - \overline{y}_{train}) \tag{30}$$

The *pseudo-R^2* criterion allows us firstly to keep those v values that present simultaneously high trading performances and higher criterion values and secondly constrain the range of the fixed values of γ, saving us a great amount of computational time. For $\gamma^{F1} \geq 1.33$ the criterion obtains values close to zero or even negative, which is evidence of over-fitting. Based on the above, we calculate the optimal $v^{F1}=0.64$. The final step is to perform again the cross-validation process for γ parameter, with $C^{F1}=0.02$ and $v^{F1}=0.64$, but also with the constraint provided by the *pseudo-R^2criterion,* thus $\gamma^{F1} \leq 1.33$. Based on this procedure, we obtain our final F1 forecast combinations with $C^{F1}=0.02$, $v^{F1}=0.54$and $\gamma^{F1}=0.63$ selected as parameters for our RBF ν-SVR model. Similarly, for F2 and F3 we obtain $C^{F2}=0.022$, $\nu^{F2}=0.71$, $\gamma^{F2}=1.41$ and $C^{F3}=0.021$, $\nu^{F3}=0.32$, $\gamma^{F3}=0.98$ respectively. Small values of γ are in general welcome because they result in smoother marginal decisions. The restrictiveness of the SVR 'tube' though depends on all three parameters and therefore it is difficult to assess if our model is more adaptive in one exercise than another. Following the same procedure we are able to collect the set of parameters for the forecasting exercises of the other two exchange rates. All sets of parameters are summarized in the following table.

[Insert Table 4]

6. STATISTICAL PERFORMANCE

As it is standard in the literature, in order to evaluate statistically our forecasts, the RMSE, the MAE, the MAPE and the Theil-U statistics are computed. For all four of the error statistics retained the lower the output, the better the forecasting accuracy of the model concerned. Their mathematical formulas are presented in Appendix D. Tables 5 summarizes the out-of-sample statistical performances of every model in each exercise for the three exchange rates.[6]

[Insert Table 5]

From the table above, we note that the SVR presents the best in-sample statistical in all out-of-sample sub-periods. All forecast combinations are statistically more accurate than the NNs. Concerning the individual models, our PSN architecture seems superior for the statistical measures retained from our individual forecasts, having a close performance with the Simple Average. RNN and MLP are following with the second and third more statistically accurate forecasts for individual models, while our RW, ARMA and STAR strategies present the less accurate in-sample forecasts for the series and periods under study. The worse realizations of the statistics are given in F2. Thus, during 2009 – 2010 our models present the worse statistical performance. This period coincides with the start of the EU debt crisis. In the next sub-period (F3), our models perform considerably better. This is happening although the EMU dept crisis is in peak

[6] The statistical in-sample results are not presented for the sake of space and they are available upon request. The statistical ranking of the models in-sample is consistent with the out-of-sample one.

and the EURO presents a volatile behavior. The trend of the results is consistent in every exchange rate under study.

In order to further verify the statistical superiority of our best proposed architecture, we compute the Modified Diebold-Mariano (MDM) statistic for forecast encompassing, as proposed by Harvey *et al.* (1997). The null hypothesis of the test is the equivalence in forecasting accuracy between a couple of forecasting models. The MDM statistic is an extension of the Diebold-Mariano (1995) test and its statistic is presented below:

$$MDM = T^{-1/2}\left[T+1-2k+T^{-1}k\left(k-1\right)\right]^{1/2} DM \tag{31}$$

where T the number of the out-of-sample observations and k the number of the step-ahead forecasts. In our case we apply the MDM test to couples of forecasts (SVR vs. another forecasting model). A negative realization of the MDM test statistic indicates that the first forecast (SVR) is more accurate than the second forecast. The lower the negative value, the more accurate are the SVR forecasts. The MDM test follows the student distribution with *T-1* degrees of freedom.

The use of MDM is common practice in forecasting because it is found to be robust in assessing the significance of observed differences between the performances of two forecasts (Barhoumi *et al.,* 2010). MDM also overcomes the problem of over-sized DMs in moderate samples (Dreger and Kholodilin, 2013). The statistic is measured in each out-of-sample period, while MSE and MAE are used as loss functions. Table 6 below presents the values of the statistics for all the cases, comparing the GA-SVR with its benchmarks.

[Insert Table 6]

From the above table it is obvious that the MDM null hypothesis of equal forecasting accuracy is rejected for all comparisons and for both loss functions at the 1% confidence interval. The statistical superiority of the SVR forecasts is confirmed as the realizations of the MDM statistic are negative for both loss functions.

7. TRADING PERFORMANCE

Further to a statistical evaluation, we evaluate our models in terms of trading efficiency. It is indeed interesting to see if their trading performance is consistent with their statistical accuracy. In section 7.1 below, we evaluate the trading performance of our models and discuss the effect of our proposed fitness function, while in section 7.2 we introduce a sophisticated trading strategy and test if its application can increase the profitability of our models.

7.1 TRADING PERFORMANCE

Our trading strategy is to go or stay 'long' when the forecast return is above zero and go or stay 'short' when the forecast return is below zero. For example, the 'long' and 'short' EUR/USD position is defined as buying and selling Euros at the current price respectively. The transaction costs for a tradable amount, say USD 5-10 million, are about 1 pip (0.0001 EUR/USD) per trade (one way) between market makers. The EUR/USD time series is considered as a series of middle rates, so the transaction costs are one spread per round trip. The average of EUR/USD is 1.421, 1.36 and 1.338 for the F1, F2 and F3 out-of-sample period respectively. Therefore, the respective costs of 1 pip are equivalent to an average cost of 0.007%, 0.0074% and 0.0075% per position. Similarly, we calculate the costs of 1 pip in the case of EUR/GBP and EUR/CHF. The trading performance measures and their calculation are presented in Appendix D. In Table 7 we present the out-of-sample trading performances[7] of our models and forecast combinations after transaction costs for each exercise and exchange rate.

[Insert Table 7]

From the above table, we note that all our NN and forecast combination models present a positive trading performance after transaction costs. From our single forecasts, the PSN outperforms each NN and statistical benchmark in terms of annualized return and information ratio. Our other two NNs architectures, the RNN and the MLP, present the second and third best trading performance respectively. This ranking is consistent in all three exercises and exchange rates under study. In the case of EUR/USD, PSN presents on

[7] Similarly with the statistical in-sample results, the in-sample trading results are not presented for the sake of space and they are available upon request. The trading ranking of the models in-sample is consistent with the out-of-sample one.

average 2.59 % higher annualized return and 0.25 higher information ratio compared to our second best single model, the RNN, for the three out-of-sample periods. All forecast combination models present improved out-of-sample trading performance, verifying a 'combining for improvement' trend. Our SVR forecast combination outperform its benchmarks and achieves on average 4.12% and 2.42% higher annualized return compared to the Kalman Filter and GP model respectively. In the trading exercise on the EUR/GBP, PSN achieve an annualized return of 13.62% and information ratio of 1.49 on average. RNN presents the second best performance with a 11.36% and 1.25 average returns and information ratio respectively. The SVR profits remain higher than the rest combining techniques, reaching up to the level of 23.27% during F3 period. Similar are the results in the case of EUR/CHF. For example, PSN's average profits are 14.89% with an average information ratio is 1.45 after transaction costs. The SVR methodology continues to present the best results, with a 2.59% on average higher profitability than our second best forecast combination technique, the GP. In terms of information ratios, PSN and SVR have on average 0.12 and 0.26 higher ones than RNN and GP respectively. In general, the trading performance of our models in F1, F2 and F3 sub-period coincides with the statistical one. The best trading results are obtained during F3 and the worst during F2. In addition to the above, we note that combining forecasts decreases the maximum drawdown, the essence of risk for an investor in financial markets.

Concerning our proposed fitness function in equation (5), the results from the statistical and trading evaluation of our individual and combining forecasts seem promising. Firstly we note that all our NNs present significant profits after transaction costs in all out-of-sample sub-periods and different exchange rates scenarios. Moreover, there are not large inconsistencies in our statistical and trading performance of our NNs models between the in-sample and out-of-sample. Large inconsistencies could indicate that the training of our NNs is biased to either statistical accuracy or trading efficiency. This could possibly lead to promising in-sample forecasts but disastrous out-of-sample results. In the next section, we introduce a trading strategy to further improve the trading performance of our models.

7.2 TRADING PERFORMANCE EXPLOITING HYBRID LEVERAGE

In order to further improve the trading performance of our models we introduce a hybrid leverage based on two time-varying factors, a leverage based on daily volatility forecasts (L1) and a leverage based on market shocks (L2). Our proposed leverage for every trading day is simply the average of L1 and L2. In the next sections we discuss how L1 and L2 are assigned.

7.2.1 Volatility Leverage (L1)

The intuition of the Volatility Leverage (L1) is to avoid trading when volatility of the exchange rate returns is very high, while at the same time exploiting days with relatively low volatility. The opposition between market-timing techniques and time-varying leverage is only apparent, as time-varying leverage can be easily achieved by scaling position sizes inversely to recent risk measures behavior.

Firstly, we forecast with a GJR (1,1)[8] the one day ahead realised volatility of each exchange rate in the test and validation sub-periods. Then, we split these two periods into six sub-periods, ranging from periods with extremely low volatility to periods experiencing extremely high volatility. Periods with different volatility levels are classified in the following way: first the average (μ) difference between the actual volatility in day t and the forecasted for day *t+1* and its 'volatility' (measured in terms of standard deviation σ) are calculated; those periods where the difference is between μ plus one σ are classified as 'Lower High Vol. Periods'. Similarly, 'Medium High Vol.' (between $\mu + \sigma$ and $\mu + 2\sigma$) and 'Extremely High Vol.' (above $\mu + 2\sigma$) periods can be defined. Periods with low volatility are also defined following the same 1σ and 2σ approach, but with a minus sign. For each sub-period a daily L1 is assigned starting with *0* for periods of extremely high volatility to a L1 of *2* for periods of extremely low volatility. Table 8 below presents the sub-periods and their relevant L1s.

[Insert Table 8]

The parameters of our strategy (μ and σ) are updated every three months by rolling forward the estimation period. So for example, for the first three months of our validation period, μ and σ are computed based on the twenty four months of the test sub-period. For the following three months, the two parameters are

[8] We also explored the RiskMetrics, GARCH (1,1) and GARCH-M models for forecasting volatility. Their statistical accuracy in all three test sub-periods is slightly worse compared with the GJR (1,1) daily volatility forecasts. Moreover, when we measure their utility in terms of trading efficiency for our models within the context of our strategy in the test sub-period, our results in terms of annualised returns are slightly better with GJR (1,1) for most of our models. The ranking of our models in terms of information ratio and annualised return is the same whether we use GJR (1,1) or the other explored alternatives. The results obtained with RiskMetrics, GARCH (1,1) and GARCH-M are available upon request.

computed based on the last twenty one months of our test sub-period and the first three of the validation sub-period.

7.2.2 Index Leverage (L2)

In the following explanation of the index leverage factor we use as an example the scenario of trading EUR/USD. The L1 measure presented above exploits periods of low volatility, but it does not take into account the effects on the EUR/USD exchange rate deriving from possible daily shocks in the EU and USA stock markets. For that reason, we introduce an Index Leverage (L2), based on two representative indices, the Dow Jones Industrial Average Index (DJIA) and the Dow Jones EuroStoxx 50 Index (SX5E). These indices efficiently reflect any shocks in the USA and European economies and are used as their proxies in a wealth of relevant studies (see amongst others Charles and Darné (2006), Tastan (2006), Hemminki and Puttonen (2008) and Awartani et *al.* (2009)).

The intuition of this leverage is to capture the shocks that our time series models are unable to incorporate in the short-run (for example the devaluation from a rating agency of an EMU country or a change in US interest rates). These changes will be instantly reflected in the affected stock market and the next day in the ECB EUR/USD fixing. However, our time series models will need some period to adjust to these shocks and are certainly unable to reflect them in the short-run.

Our methodology is similar to the one followed for L1. Firstly, we define the daily difference $\delta_{E\text{-}U}$ as:

$$\delta_{E-U} = R_{SX5E} - R_{DJIA} \qquad (32)$$

where R_{SX5E} and R_{DJIA} are the daily SX5E and DJIA stock index returns respectively.[9] We also compute the mean of that difference (μ') and its standard deviation (σ'). Then based on $\delta_{E\text{-}U}$, μ' and σ', we split every three months of the test and the out-of-sample into six sub-periods. The parameters of our strategy (μ' and σ') are again updated every three months by rolling forward the estimation period. The sub-periods are generated as in L1. Namely, the periods where the difference $\delta_{E\text{-}U}$ is between μ' plus one σ' are classified as

[9] Regarding EUR/USD case, DJIA's closing time is at 4:30 a.m. (ECT), while SX5E is closing at 6:00 p.m. (ECT). Since ECB's daily fixing is available at 2.15 p.m. (ECT), we calculate today's $\delta_{E\text{-}U}$ with the first lags of the stock index returns R_{SX5E} and R_{DJIA}. We use the same calculation for the cases of EUR/GBP and EUR/CHF. All stock index returns are calculated as in equation (1).

'Lower High δ_{E-U} Periods'. Similarly, 'Medium High δ_{E-U}' (between $\mu'+\sigma'$ and $\mu'+2\sigma'$) and 'Extremely High δ_{E-U}' (above $\mu'+2\sigma'$) periods can be defined. Periods with low difference δ_{E-U} are also defined following the same $1\sigma'$ and $2\sigma'$ approach, but with a minus sign. When δ_{E-U} is considerably higher than the average (a positive shock in the euro zone), we should expect that the EUR will appreciate.

In order to justify the application of our leverage though, we further separate the trading days based on the sign of the daily forecast. Thus, we have the following two scenarios:

- If the sign of the forecast is positive (we are 'long'), we apply a leverage ($L2^{+}$) of more than 1.
- If the sign of the forecast is negative (we are 'short'), we apply a leverage ($L2^{-}$) of less than 1.

When the δ_{E-U} is considerably lower than the average (a negative shock in the euro zone), we should expect a depreciation of the EUR. Thus, the assigned leverage has the opposite trend in the corresponding scenario. In the same way, we obtain the δ's and assign the leverage factors for the cases of EUR/GBP and EUR/CHF. The FTSE 100 index and Swiss Market Index (SMI) reflect the UK and Swiss Stock Exchange respectively. Therefore, they are used as proxies of the UK and Swiss markets, potentially reflecting shocks in the state of the economy of these two countries (Charles and Darné (2006) and Äijö (2008)). The SX5E index is still used as a 'detector' of the European shocks, the general classification of L2 is shown in Table 9 below.

[Insert Table 9]

7.2.3 Hybrid Leverage Performance

From the above, the L1 and L2 (depending on the scenario, $L2^{+}$ or $L2^{-}$) factors are available for each trading day of the three exchange rates. We apply to each individual model a daily hybrid leverage equal to the simple average of L1 and L2 and check if its trading performance improves. A depiction of these leverage factors through the periods F1, F2 and F3 is given in the following figure. These assigned factors refer to our model with the best statistical and trading performance, the SVR, for the case of EUR/USD.

[Insert Figure 4]

From the figure above, we note that the volatility based leverage (L1) takes mainly low values during 2008, through the F2 and the first semester of the F3 sub-period. Regarding L2, the trend is more irregular and in general more extreme, going from very low values to high ones in short period intervals. This can be attributed to the economic turbulence that dominates the out-of-sample periods and the shocks in the two benchmark markets. The L1 and L2 graphs for our other NN and forecast combination models present similar behaviors for the three periods and the three exchange rates under study, but are not presented in this paper for the sake of space.

The cost of leverage (interest payments for the additional capital) is calculated at 0.504% p.a. (that is 0.002% per trading day[10]). Our final results are presented in Table 10 below[11].

[Insert Table 10]

Based on the results presented above, we can argue that our hybrid trading strategy is successful for all our models and periods. In the case of EUR/USD, the SVR forecast combination seems to exploit our trading strategy well and achieves on average an annualized return of 25.08% after costs, increasing its profitability by 3.05%, 4.16% and 2.05% during F1, F2 and F3 sub-periods respectively. GP and Kalman Filter remain the second and the third most profitable models, achieving both on average annualized returns over 20%, regardless the period under study. In general all forecasting models increase their trading performance by 1.69% on average while their maximum drawdown is decreased on average by 1.01%. The obtained results are similar for EUR/GBP. The SVR average profits are on average 23.31%, while its annualized returns are on average increased by 2.35% thought out the three different out-of-samples. The average profitability of GP and Kalman reaches the level of 20.9% and 19.39% respectively. The average increase of trading performance is around 2%. On the other hand, the maximum drawdown decreases on average by 0.90%. Finally, the EUR/CHF case suggests that our best model's trading efficiency is again confirmed by increasing its profitability after leverage by 1.66%, 1.91% and 3.13% in F1, F2 and F3

[10] The interest costs are calculated by considering a 0.504% interest rate p.a. (the Euribor rate at the time of calculation) divided by 252 trading days. In reality, leverage costs also apply during non-trading days so that we should calculate the interest costs using 360 days per year. But for the sake of simplicity, we use the approximation of 252 trading days to spread the leverage costs of non-trading days equally over the trading days. This approximation prevents us from keeping track of how many non-trading days we hold a position.

[11] The hit ratios are not affected with the application of the leverage and therefore are not presented here in order to avoid repetition.

respectively. GP achieves on average annualized returns of 21.39%, but still it is outperformed by SVR by 2.21%. Kalman Filter is the third best model once more, but it does achieve high profits, especially in F3. The average profit increase of the models after leverage is 1.62%, while the average maximum drawdown decrease is 0.55%. In sub-periods F1, F2 and F3 the three exchange rates are dominated by shocks and high volatility (see figure 1). Our leverage factors manage to exploit this environment and increase the trading performance of all our models, in periods where uncertainty is present in the market. When analyzing the different out-of-sample periods for EUR/USD, the average of the annualized returns of all the models is 13.58%. The respective results for the EUR/GBP and EUR/CHF are 13.36% and 13.84%. This proves that FX profitability can be achieved, while the EUR/CHF trading seems to be marginally more opportunistic.

8. CONCLUSIONS

In this paper we examine the performance of a MLP, RNN and PSN architecture in forecasting and trading the EUR/USD, EUR/GBP and EUR/CHF. The utility of Kalman Filter, GP and SVR algorithms as forecasting combination techniques is also explored. As benchmarks for our NNs we use a RW, an ARMA and a STAR, while for our forecast combination techniques a Simple Average and a LASSO. We also introduce a new fitness function for NNs in trading applications and a hybrid leverage trading strategy, in order to evaluate if their application can improve the trading performance of our models.

In terms of our results, the PSN from the individual forecasts and the SVR from our forecasting combination techniques outperform their benchmarks in terms of statistical accuracy and trading efficiency. All NN forecast combinations achieve higher annualized returns and information ratios, presenting a 'combining for improvement' pattern. Concerning our hybrid leverage strategy, we note that all models exploit it by increasing annualized returns and decreasing maximum drawdowns. The hybrid leverage factors applied serve their purpose, since they are more effective in periods of increased market volatility and risk. Moreover, our proposed fitness function for NNs is promising as all networks produce high profitability and present a consistency between their statistical and trading ranking. It is also observed that the ranking of all models is consistent in statistical and trading terms. Finally, the robustness of our results is verified through three rolling forecasting exercises, which embody in- and out-of-sample periods of economic turmoil.

The remarkable trading performance of the SVR allows us to conclude that it can be considered as the optimal forecasting combination for the models and time-series under study. The successful application of our proposed trading strategy and fitness function demonstrates the necessity for a shift from purely statistically based models to models that are optimized in a hybrid trading and statistical approach.

APPENDIX

A. Descriptive Statistics

The following figure summarizes the descriptive statistics of EUR/USD, EUR/GBP and EUR/CHF.

[Insert Figure A.1]

All series are found to be non-normal at 99% confidence interval, presenting skewness and kurtosis. The stationarity of our return series is also confirmed at 99% confidence by the ADF statistics, which are presented in the following table.

[Insert Table A.1]

B. NNs' Training Characteristics

In Table B.1 we present the characteristics of the NNs with the best trading performance in the in-sample sub-period, which we used in our committees. The choice of these parameters is based on sensitivity tests in the in-sample sub-period and on the relevant literature (Tenti (1996), Zhang *et al.* (1998) and Ghazali *et al.* (2006)).

[Insert Table B.1]

C. Genetic Programming Characteristics

Table C.1 presents the parameters selected in our GP application.

[Insert Table C.1]

D. Statistical and Trading Performance Measures

The statistical and trading performance measures are calculated as shown in table D.1 and table D.2 respectively:

[*Insert Table D.1 and Insert Table D.2]*

REFERENCES

Äijö, J., 2008. Impact of US and UK macroeconomic news announcements on the return distribution implied by FTSE-100 index options. International Review of Financial Analysis 17, 242-258.

Altavilla, C., De Grauwe, P., 2008. Forecasting and combining competing models of exchange rate determination, Applied Economics 42, 3455-3480.

Andersen, T.G., Bollerslev, T., Diebold, F.X., Vega, C., 2003. Micro Effects of Macro Announcements: Real-Time Price Discovery in Foreign Exchange. American Economic Review 93, 38-62.

Andrawis, R.R., Atiya, A.F., El-Shishiny, H., 2011. Forecast combinations of computational intelligence and linear models for the NN5 time series forecasting competition. International Journal of Forecasting 27, 672–688.

Awartani, B., Corradi, V., Distaso, W., 2009. Assessing Market Microstructure Effects via Realized Volatility Measures with an Application to the Dow Jones Industrial Average Stocks. Journal of Business and Economic Statistics 27, 251-265.

Bacchetta, P., Wincoop, E.V., 2013. On the unstable relationship between exchange rates and macroeconomic fundamentals. Journal of International Economics 91, 18-26.

Bahrammirzaee, A., 2010. A comparative survey of artificial intelligence applications in finance: artificial neural networks, expert system and hybrid intelligent systems. Neural Computing and Applications 19, 1165-1195.

Barhoumi, K., Darné, O., Ferrara, L., 2010. Are disaggregate data useful for factor analysis in forecasting French GDP? Journal of Forecasting 29, 132–144.

Basak, D., Pal, S., Patranabis, D. C., 2007. Support Vector Regression. International Journal of Neural Information Processing 10, 203-224.

Bates, J.M., Granger, C.W.J., 1969. The Combination of Forecasts. Journal of the Operational Research Society 20, 451-468.

Bissoondeeal, R.K., Binner, J.M., Bhuruth, M., Gazely, A., Mootanah, V.P., 2008. Forecasting exchange rates with linear and nonlinear models. Global Business and Economics Review 10, 414-429.

Brooks, C., 2008. Introductory econometrics for finance, 2nd Ed. Cambridge, University Press.

Cao, L.J., Chua, K.S., Guan, L.K., 2003. C-ascending support vector machines for financial time series forecasting. In: Computational Intelligence for Financial Engineering Proceedings, 317-323.

Chalimourda, A., Schölkopf, B., Smola, A., 2004. Experimentally optimal ν in support vector regression for different noise models and parameter settings. Neural Networks 17, 127-141.

Chan, Y. L., Stock, J. H., Watson, M.W., 1999. A dynamic factor model framework for forecast combination. Spanish Economic Review 1, 91-121.

Chan, K.S., Tong, H., 1986. On estimating thresholds in autoregressive models. Journal of Time Series Analysis 7, 178-190.

Charles, A, Darné, O., 2006. Large shocks and the September 11th terrorist attacks on international stock markets. Economic Modeling 23, 683-698.

Chen, S.H., 2002. Genetic algorithms and genetic programming in computational finance. Boston: Kluwer Academic Publishers.

Chen, A.S., Leung, M.T., 2004. Regression neural network for error correction in foreign exchange forecasting and trading. Computers and Operations Research 31, 1049-1068.

Cherkassky, V., Ma, Y., 2004. Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks 17, 113-126.

De Gooijer, J.G., Hyndman, R.J., 2006. 25 years of time series forecasting. International Journal of Forecasting 22, 443-473.

De Menezes, L.M., Nikolaev, N.Y., 2006. Forecasting with genetically programmed polynomial neural networks. International Journal of Forecasting 22, 249-265.

Diebold, F.X., Mariano, R.S., 1995. Comparing Predictive Accuracy. Journal of Business and Economic Statistics 13, 253-263.

Donaldson, R.G., Kamstra, M., 1999. Neural network forecast combining with interaction effects. Journal of the Franklin Institute 336, 227-236.

Dreger, C., Kholodilin, K.A., 2013. Forecasting Private Consumption by Consumer Surveys. Journal of Forecasting 32, 10–18.

Duan, K, Keerthi, S.S., Poo, A.N., 2003. Evaluation of simple performance measures for tuning SVM hyperparameters. Neurocomputing 51, 41-59.

Dueker, M., Neely, C.J., 2007. Can Markov switching models predict excess foreign exchange returns?. Journal of Banking and Finance 31, 279-296.

Dunis, C. L., Laws, J., Sermpinis, G., 2010. Modelling and trading the EUR/USD exchange rate at the ECB fixing. The European Journal of Finance 16, 541-560.

Dunis, C.L., Laws, J., Sermpinis, G., 2011. Higher order and recurrent neural architectures for trading the EUR/USD exchange rate. Quantitative Finance 11, 615-629.

Ebrahimpour, R., Nikoo, H., Masoudnia, S., Yousefi, M.R., Ghaemi, M.S., 2011. Mixture of MLP-experts for trend forecasting of time series: A case study of the Tehran stock exchange. International Journal of Forecasting 27, 804-816.

Evans, K.P., Speight, A.E.H., 2010. Dynamic news effects in high frequency Euro exchange rates. Journal of International Financial Markets, Institutions and Money 20, 238–258.

Fernández-Rodríguez, F., Sosvilla-Rivero, S., Andrada- Félix, J., 2003. Technical analysis in foreign exchange markets: evidence from the EMS. Applied Financial Economics 13, 113–122.

Flannery, M.J., Protopapadakis, A.A., 2002. Macroeconomic Factors Do Influence Aggregate Stock Returns. Review of Financial Studies 15, 751-782.

Ghazali, R., Hussain, A., Merabti, M., 2006. Higher Order Neural Networks for Financial Time Series Prediction. The 10th IASTED International Conference on Artificial Intelligence and Soft Computing 1, 119-124.

Grossmann, A., McMillan, D.G., 2010. Forecasting exchange rates: Non-linear adjustment and time-varying equilibrium. Journal of International Financial Markets, Institutions and Money 20, 436-450.

Guidolin, M., Timmermann, A., 2009. Forecasts of US short-term interest rates: A flexible forecast combination approach, Journal of Econometrics 150, 297-311.

Hansen, P.R., 2005. A test for superior predictive ability. Journal of Business and Economic Statistics 23, 365–380.

Harrald, P.G., Kamstra, M., 1997. Evolving artificial neural networks to combine financial forecasts. IEEE Transactions on Evolutionary Computation 1, 40-52.

Harvey, D., Leybourne, S., Newbold, P., 1997. Testing the equality of prediction mean squared errors. International Journal of Forecasting 13, 281–291.

Hastie, T., Tibshirani, R., Friedman, J.H., 2009. The elements of statistical learning : data mining, inference, and prediction, 2nd Ed. New York, Springer.

Hemminki, J., Puttonen, V., 2008. Fundamental indexation in Europe. Journal of Asset Management 8, 401-405.

Hu, M.Y. Tsoukalas, C., 1999. Combining conditional volatility forecasts using neural networks: an application to the EMS exchange rates. Journal of International Financial Markets, Institutions and Money 9, 407–422.

Ince, H., Trafalis, T.B., 2006. Kernel methods for short-term portfolio management. Expert Systems with Applications 30, 535-542.

Ince, H., Trafalis, T.B., 2008. Short term forecasting with support vector machines and application to stock price prediction. International Journal of General Systems 37, 677-687.

James, J., Marsh, I., Sarno, L., 2012. Handbook of Exchange Rates. Chicago,Wiley.

Jarque, C.M., Bera, A.K., 1980. Efficient tests for normality, homoscedasticity and serial independence of regression residuals. Economic Letters 6, 255-259.

Koza, J.R. (1992), *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, Cambridge MIT Press

Kröse, B., Smagt, P.V.D., 1996. An introduction to neural networks. Eight ed. University of Amsterdam.

Lam, M., 2004. Neural network techniques for financial performance prediction, integrating fundamental and technical analysis. Decision Support Systems 37, 567–581.

Lee, C.I., Mathur, I., 1996. Trading rule profits in European currency spot cross-rates. Journal of Banking and Finance 20, 949–962.

Lin, C.J., Teräsvirta T., 1994. Testing the Constancy of Regression Parameters against Continuous Structural Changes. Journal of Econometrics 62, 211-228.

Ling, S.H., Leung, F.H.F., Lam, H.K., Yim-Shu, L., Tam, P.K.S., 2003. A novel genetic-algorithm-based neural network for short-term load forecasting. IEEE Transactions on Industrial Electronics 50, 793-799.

Lo, A., 2004. The adaptive markets hypothesis. The Journal of Portfolio Management 30, 15-29.

Lu, C.J., Lee, T.S., Chiu, C.C., 2009. Financial time series forecasting using independent component analysis and support vector regression. Decision Support Systems 47, 115-125.

Meese, R.A., Rogoff, K., 1983a. Empirical exchange rate models of the seventies: Do they fit out of sample? Journal of International Economics 14, 3–24.

Meese, R.A., Rogoff, K., 1983b. The out-of-sample failure of empirical exchange rate models: sampling error or misspecification? In: Frankel, J.A. (Ed) Exchange rates and international macroeconomics, University of Chicago Press, 67-112.

Montana, G., Parrella, F., 2008. Learning to Trade with Incremental Support Vector Regression Experts Hybrid Artificial Intelligence Systems. In: Corchado, E., Abraham, A., Pedrycz, W. (Eds.), Hybrid Artificial Intelligence Systems, Springer Berlin, Heidelberg, 591-598.

Neely, C.J., 2002. The temporal pattern of trading rule returns and exchange rate intervention: intervention does not generate technical trading profits. Journal of International Economics 58, 211-232.

Paliwal, M., Kumar, U.A., 2009. Neural networks and statistical techniques: A review of applications. Expert Systems with Applications 36, 2–17.

Park, C.H., Irwin, S.H., 2007. What do we know about profitability of technical analysis?. Journal of Economic Surveys 21, 786–826.

Pearce, D.K., Solakoglu M.N., 2007. Macroeconomic news and exchange rates. Journal of International Financial Markets, Institutions and Money 17, 307–325.

Poli ,R., McPhee, N. 2008. Parsimony pressure made easy, *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, Atlanta, GA, USA

Qi, M., Wu, Y., 2006. Technical Trading-Rule Profitability, Data Snooping, and Reality Check: Evidence from the Foreign Exchange Market. Journal of Money, Credit and Banking 38, 2135-2158.

Rozeff, M., Kinney, W., 1976. Capital Market Seasonality: The Case of Stock Returns. Journal of Financial Economics 3, 379-402.

Schölkopf, B., Bartlett, P., Smola, A., Williamson, R., 1999. Shrinking the tube: a new support vector regression algorithm. In: Kearns, M. J. (Ed.), Advances in neural information processing systems 11. Cambridge: Mass, MIT Press, 330-336.

Sermpinis, G., Dunis, C., Laws, J., Stasinakis, C., 2012. Forecasting and trading the EUR/USD exchange rate with stochastic neural network combination and time-varying leverage. Decision Support Systems 54, 316-329.

Shapiro, A.F., 2000. A Hitchhiker's guide to the techniques of adaptive nonlinear models. Insurance: Mathematics and Economics 26, 119-132.

Shin, Y., Ghosh, J., 1991. The Pi-Sigma Network: An efficient Higher-order Neural Networks for Pattern Classification and Function Approximation. Proceedings of International Joint Conference of Neural Networks 1, 13-18.

Smola, A., Murata, N., Schölkopf B., Muller, K.R., 1998. Asymptotically optimal choice of ε-loss for support vector machines. In: Niklasson, L., Boden, M., Ziemke, T. (Eds), Proceedings of the International Conference on Artificial Neural Networks, Perspectives in Neural Computing. Berlin, Springer, 105–110.

Smola, A., Schölkopf, B., 2004. A tutorial on support vector regression. Statistics and Computing 14, 199-222.

Sundberg, R., 2006. Shrinkage Regression. In: El-Shaarawi, A.H., Piergosh, W.W.(eds.), Encyclopedia of Environmetrics, John Wiley and Sons Ltd.. Chichester,1994-1998.

Suykens, J.A.K., Brabanter, J. D., Lukas, L., Vandewalle, L., 2002. Weighted least squares support vector machines: robustness and sparse approximation Neurocomputing 48, 85-105.

Tastan, H., 2006. Estimating time-varying conditional correlations between stock and foreign exchange markets. Physica A. 360, 445-458.

Tenti, P., 1996. Forecasting foreign exchange rates using recurrent neural networks. Applied Artificial Intelligence 10, 567-581.

Teräsvirta, T., Van Dijk, D., Medeiros, M.C., 2005. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. International Journal of Forecasting 21, 755-774.

Tibshirani, R., 2011. Regression shrinkage and selection via the lasso: a retrospective. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73, 273-282.

Trafalis, T.B., Ince, H., 2000. Support vector machine for regression and applications to financial forecasting, Proceedings of International Joint Conference on Neural Networks 1, 348-353.

Vapnik, V.N., 1995. The nature of statistical learning theory, Springer.

Veall, M.R., Zimmermann, K.F., 1996. Pseudo-R2 measures for some common limited dependent variable models. Journal of Economic Surveys 10, 241-259.

Vellido, A., Lisboa, P.J.G, Vaughan, J. 1999. Neural networks in business: a survey of applications (1992–1998). Expert Systems with Applications 17, 51–70.

Wang, H., Li, G., Jiang, G., 2007. Robust Regression Shrinkage and Consistent Variable Selection Through the LAD-Lasso. Journal of Business and Economic Statistics 25, 347-355.

Welch, G., Bishop, G., 2001. An Introduction to the Kalman Filter. Design 7, 1-16.

Yang, Y., 2004. Combining Forecasting Procedures: Some theoretical results. Econometric Theory 20, 176-222.

Yuan, M., Lin, Y., 2007. On the non-negative garrotte estimator Journal of the Royal Statistical Society: Series B (Statistical Methodology) 69, 143-161.

Zhang, G., Patuwo, B.E., Hu, M.Y., 1998. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting 14, 35-62

Zhu, H., Rohwer, R., 1996. No free lunch for cross-validation, Neural Computation 8, 1421-1426.

FIGURES

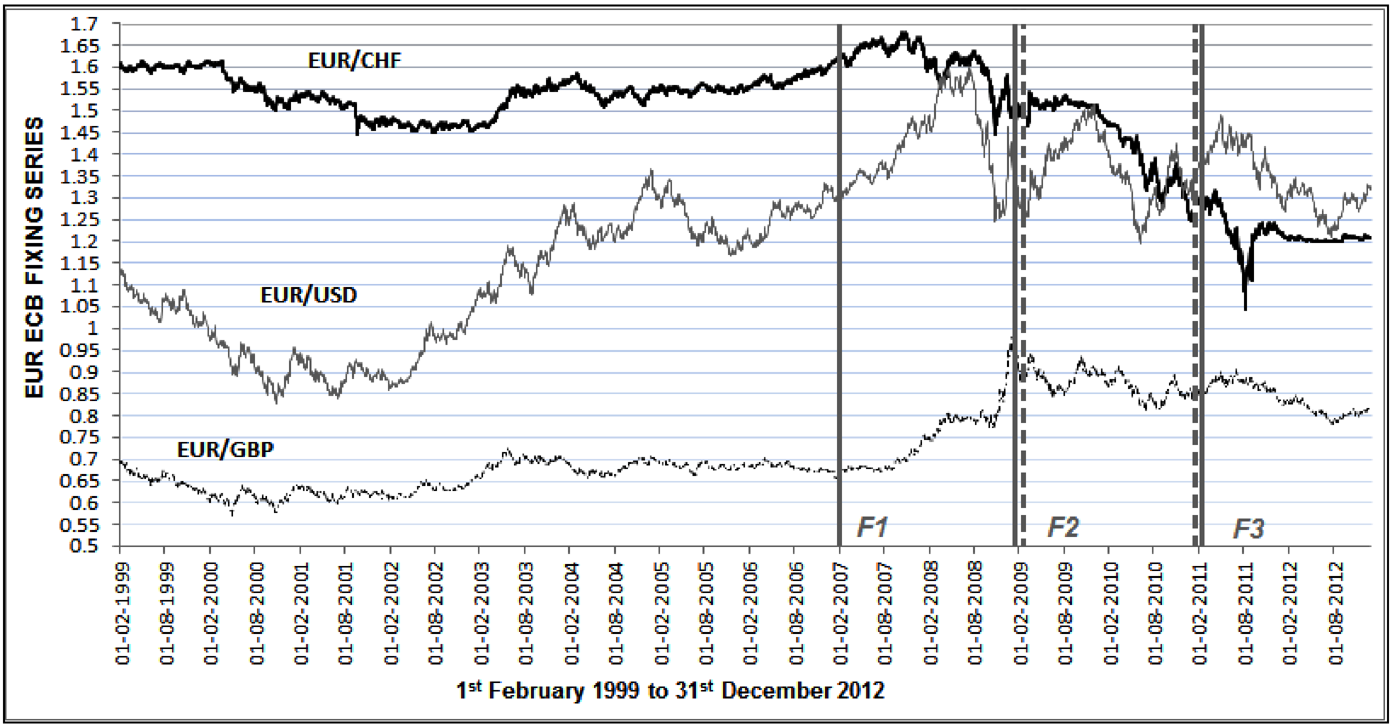
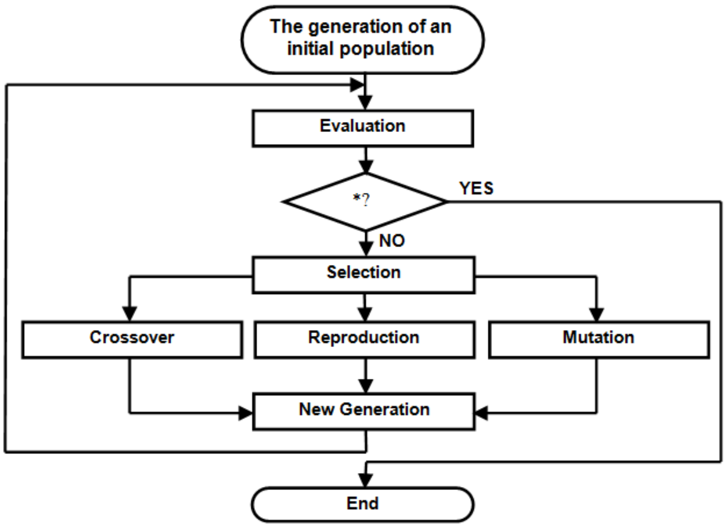


Figure 1: The three exchange rates and out-of-sample periods under study.



** The symbol '?' is the termination criterion which iterates or terminates the procedure of GP.*

Figure 2: GP Architecture

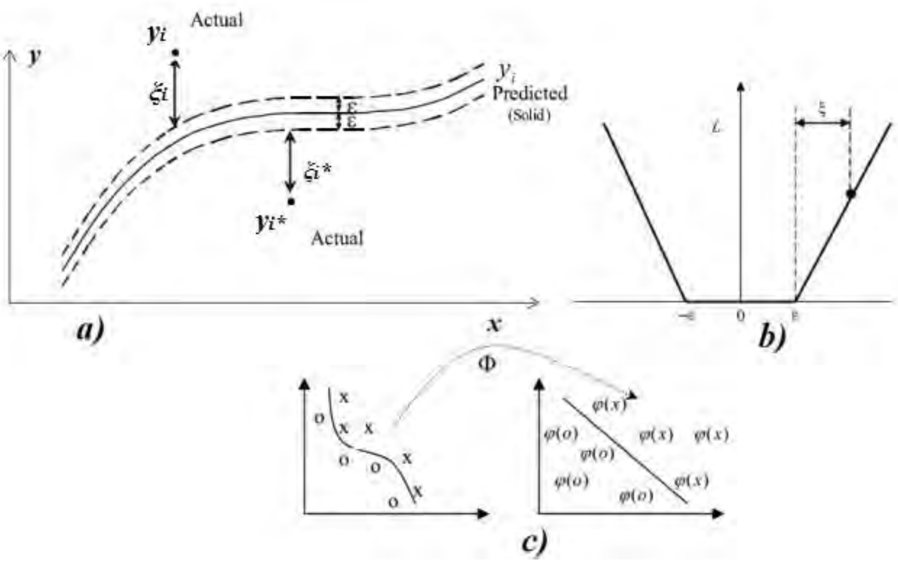


Figure 3: a) The f(x) curve of SVR and the ε-tube, b) plot of the ε-sensitive loss function and c) mapping procedure by φ*(x)*

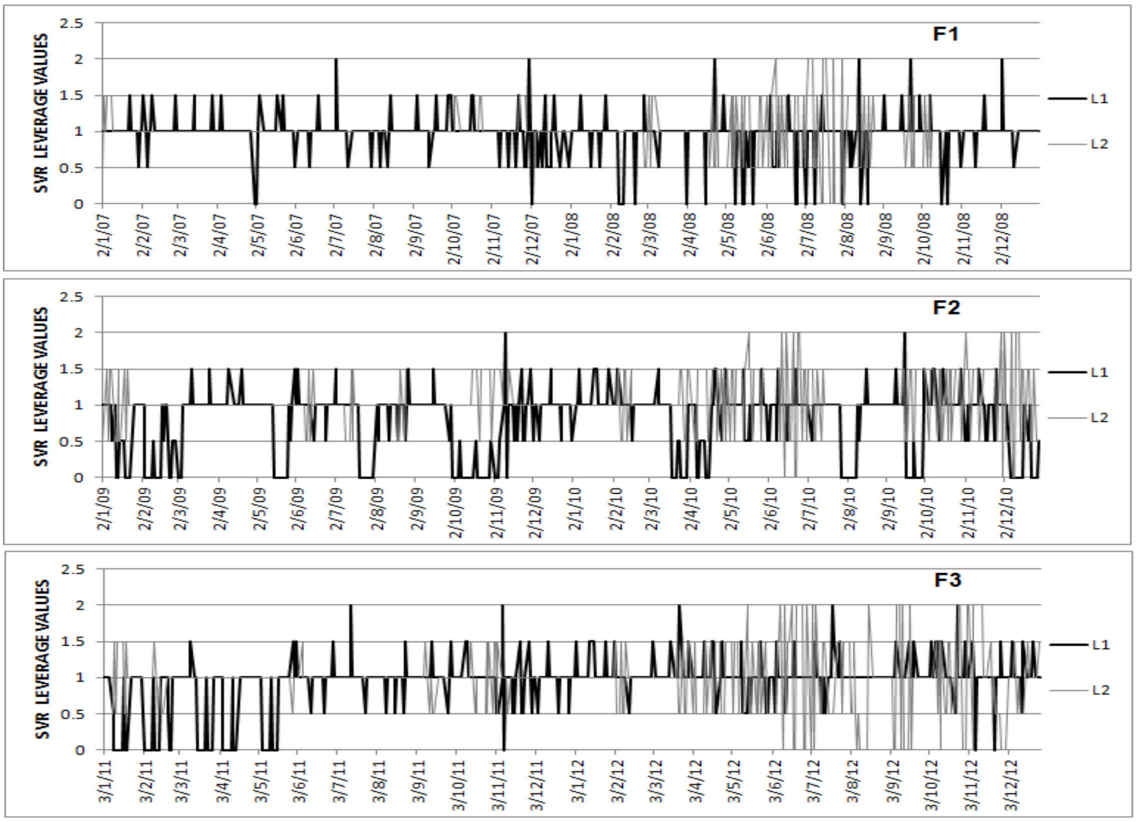


Figure 4: The Volatility Leverage (L1) and Index Leverage (L2) values assigned to the SVR model for each period under study regarding EUR/USD

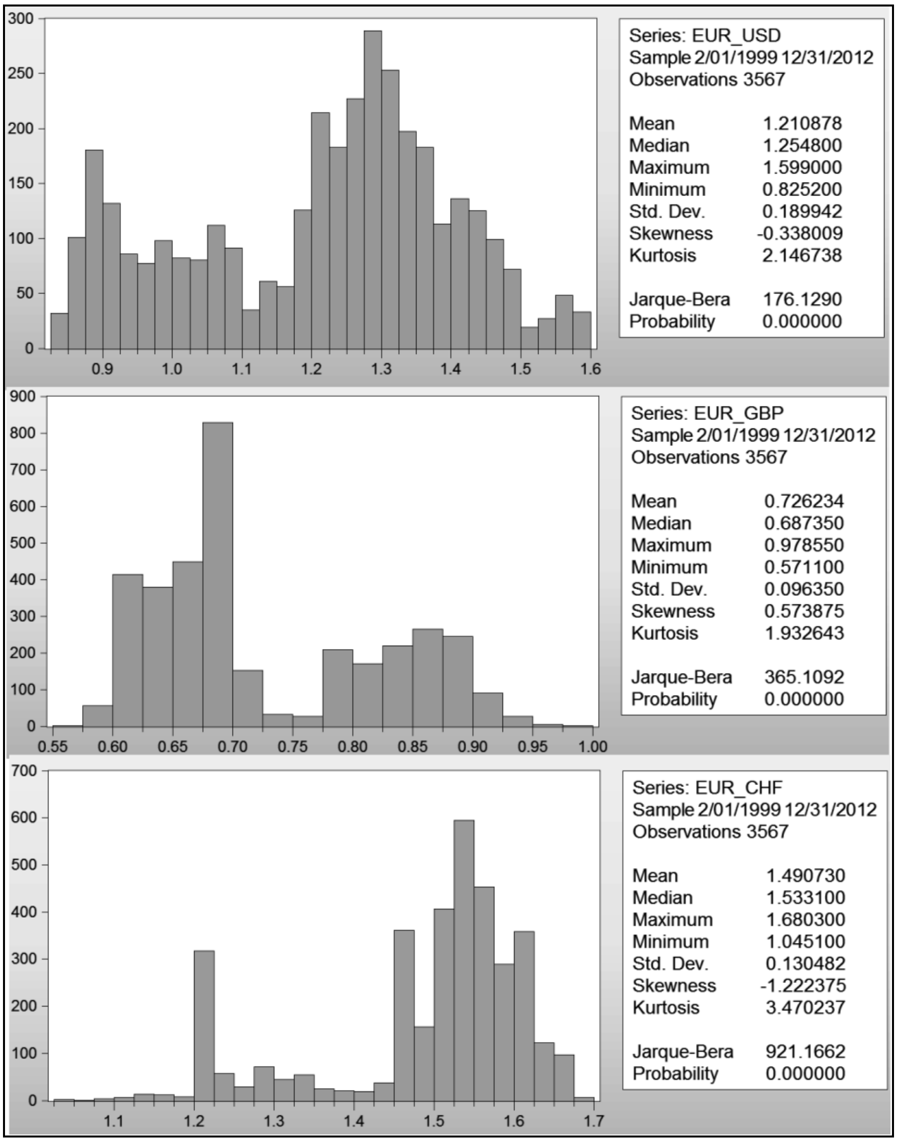


Figure A.1: Summary of descriptive statistics of the three exchange rates under study

TABLES

	F1		F2		F3	
PERIODS	**DAYS**	**PERIOD**	**DAYS**	**PERIOD**	**DAYS**	**PERIOD**
Total Dataset	2540	01/02/1999 - 31/12/2008	2560	02/01/2001 - 31/12/2010	2564	02/01/2003 - 31/12/2012
Training Dataset ***(In-sample)***	1517	01/02/1999 - 31/12/2004	1535	02/01/2001 - 29/12/2006	1537	02/01/2003 - 31/12/2008
Test Dataset ***(In-sample)***	512	03/01/2005 -29/12/2006	511	02/01/2007 -31/12/2008	514	02/01/2009 -31/12/2010
Validation Dataset ***(Out-of-sample)***	511	02/01/2007 -31/12/2008	514	02/01/2009 -31/12/2010	513	03/1/2011 -31/12/2012

Table 1: The EUR/USD Dataset and Neural Networks' Training Sub-periods for the three forecasting exercises

	EUR/USD	EUR/GBP	EUR/CHF
F1	AR (1, 2, 6) and MA (1, 2, 5)	AR (1, 4, 7, 8) and MA (2, 3, 6)	AR (1, 2, 5, 7) and MA (2, 5, 6, 8)
F2	AR (1, 4, 5, 7) and MA (1, 3, 7)	AR (1, 4, 8) and MA (1, 3, 5, 6)	AR (1, 2, 5, 7) and MA (4, 6, 7)
F3	AR (2, 3, 5, 8) and MA (1, 3, 6, 8)	AR (1, 3, 4, 6, 7) and MA (1, 2, 4)	AR (1, 2, 6) and MA (1, 3, 4)

Note: The numbers in the parentheses correspond to the lags of the AR and MA terms of each ARMA structure

Table 2: The ARMA structures used for each exchange rate and forecasting exercise

	MLP	Lags*			RNN	Lags			PSN	Lags		
	Explanatory Variables	**F1**	**F2**	**F3**	**Explanatory Variables**	**F1**	**F2**	**F3**	**Explanatory Variables**	**F1**	**F2**	**F3**
EUR/USD	EUR/USD Exch. Rate	1	1	1	EUR/USD Exch. Rate	1	1	2	EUR/USD Exch. Rate	1	3	1
	EUR/USD Exch. Rate	3	4	2	EUR/USD Exch. Rate	2	3	3	EUR/USD Exch. Rate	4	4	2
	EUR/USD Exch. Rate	5	5	5	EUR/USD Exch. Rate	3	5	6	EUR/USD Exch. Rate	7	5	5
	EUR/USD Exch. Rate	9	6	8	EUR/USD Exch. Rate	6	7	8	EUR/USD Exch. Rate	8	7	6
	EUR/USD Exch. Rate	10	7	9	EUR/USD Exch. Rate	9	11	11	EUR/USD Exch. Rate	9	9	8
	EUR/USD Exch. Rate	11	8	12	EUR/USD Exch. Rate	10	12	-	EUR/USD Exch. Rate	11	10	11
	EUR/GBP Exch. Rate	2	1	1	EUR/GBP Exch. Rate	1	1	2	EUR/GBP Exch. Rate	-	2	2
	EUR/GBP Exch. Rate	4	3	-	EUR/GBP Exch. Rate	4	4	-	EUR/JPY Exch. Rate	4	1	-
	EUR/JPY Exch. Rate	-	2	-	EUR/JPY Exch. Rate	1	3	2	EUR/JPY Exch. Rate	1	-	-
EUR/GBP	EUR/ GBP Exch. Rate	1	1	2	EUR/ GBP Exch. Rate	2	2	1	EUR/ GBP Exch. Rate	1	1	1
	EUR/GBP Exch. Rate	2	2	3	EUR/GBP Exch. Rate	3	4	2	EUR/GBP Exch. Rate	2	2	4
	EUR/GBP Exch. Rate	3	4	4	EUR/GBP Exch. Rate	5	5	5	EUR/GBP Exch. Rate	5	5	5
	EUR/GBP Exch. Rate	4	5	6	EUR/GBP Exch. Rate	6	7	7	EUR/GBP Exch. Rate	6	7	7
	EUR/GBP Exch. Rate	7	8	10	EUR/GBP Exch. Rate	8	8	12	EUR/GBP Exch. Rate	10	9	9
	EUR/GPB Exch. Rate	10	9	-	EUR/GPB Exch. Rate	11	-	-	EUR/GPB Exch. Rate	-	-	12
	EUR/USD Exch. Rate	2	1	1	EUR/USD Exch. Rate	3	3	1	EUR/USD Exch. Rate	-	3	4
	EUR/USD Exch. Rate	3	4	-	EUR/USD Exch. Rate	6	-	2	EUR/USD Exch. Rate	2	-	6
	EUR/CHF Exch. Rate	1	1	-	EUR/CHF Exch. Rate	2	1	-	EUR/CHF Exch. Rate	3	3	1
EUR/CHF	EUR/CHF Exch. Rate	2	1	3	EUR/CHF Exch. Rate	1	1	2	EUR/CHF Exch. Rate	2	1	3
	EUR/CHF Exch. Rate	3	3	6	EUR/CHF Exch. Rate	3	2	4	EUR/CHF Exch. Rate	4	4	4
	EUR/CHF Exch. Rate	5	5	9	EUR/CHF Exch. Rate	4	5	5	EUR/CHF Exch. Rate	5	6	5
	EUR/CHF Exch. Rate	6	8	10	EUR/CHF Exch. Rate	6	7	8	EUR/CHF Exch. Rate	8	7	8
	EUR/CHF Exch. Rate	7	11	11	EUR/CHF Exch. Rate	11	8	9	EUR/CHF Exch. Rate	11	9	10
	EUR/CHF Exch. Rate	-	-	-	EUR/CHF Exch. Rate	12	9	-	EUR/CHF Exch. Rate	12	10	11
	EUR/CHF Exch. Rate	-	-	-	EUR/CHF Exch. Rate	-	11	-	EUR/CHF Exch. Rate	-	-	12
	EUR/USD Exch. Rate	1	-	-	EUR/USD Exch. Rate	-	1	3	EUR/USD Exch. Rate	2	2	-
	EUR/GBP Exch. Rate	3	6	5	EUR/GBP Exch. Rate	2	-	5	EUR/GBP Exch. Rate	-	3	2

**In our case the term 'Lag 1' means that today's closing price is used to forecast tomorrow's one. F1, F2 and F3 columns present the lags selected for every NN in each forecasting exercise.*

Table 3: Explanatory variables for each NN model and exchange rate

	F1			F2			F3		
	EUR/USD	EUR/GBP	EUR/CHF	EUR/USD	EUR/GBP	EUR/CHF	EUR/USD	EUR/GBP	EUR/CHF
C	0.02	0.15	0.06	0.022	0.09	0.08	0.021	0.042	0.102
v	0.54	0.68	0.47	0.71	0.62	0.57	0.32	0.23	0.79
γ	0.63	0.91	0.84	1.41	1.36	1.52	0.98	1.05	1.41

Table 4: SVR parameters for each application

			TRADITIONAL STRATEGIES			NEURAL NETWORKS			FORECAST COMBINATIONS				
OUT-OF-SAMPLE			RW	ARMA	STAR	MLP	RNN	PSN	AVERAGE	LASSO	KALMAN	GP	SVR
EUR/USD	F1	MAE	0.0081	0.0064	0.006	0.0058	0.0056	0.0053	0.0052	0.0047	0.0046	0.0043	0.0039
		MAPE	221.18%	121.18%	116.23%	106.87%	104.25%	101.28%	98.37%	95.71%	92.49%	89.54%	86.67%
		RMSE	0.0094	0.0077	0.0075	0.0074	0.0072	0.0069	0.0066	0.0063	0.0061	0.0057	0.0053
		THEIL-U	0.8867	0.8499	0.7854	0.7578	0.7519	0.7226	0.6951	0.6795	0.6732	0.6429	0.6117
	F2	MAE	0.0096	0.0069	0.0065	0.0063	0.0061	0.0059	0.0059	0.0056	0.0055	0.0052	0.0048
		MAPE	234.17%	128.44%	121.76%	107.48%	105.37%	103.72%	101.56%	99.27%	98.13%	95.27%	92.84%
		RMSE	0.0152	0.0083	0.0081	0.0074	0.0072	0.007	0.0069	0.0066	0.0064	0.0061	0.0058
		THEIL-U	0.9815	0.9301	0.8654	0.7972	0.7895	0.7664	0.7351	0.7005	0.6886	0.6758	0.6328
	F3	MAE	0.0079	0.0063	0.0059	0.0051	0.0049	0.0048	0.0047	0.0045	0.0043	0.0041	0.0037
		MAPE	186.21%	114.10%	111.18%	99.52%	98.06%	96.84%	95.73%	93.12%	89.57%	87.33%	85.27%
		RMSE	0.0086	0.0067	0.0074	0.0066	0.0065	0.0064	0.0061	0.0058	0.0055	0.0053	0.0051
		THEIL-U	0.8358	0.8194	0.7059	0.6529	0.6458	0.6297	0.6218	0.6014	0.5788	0.5617	0.5419
EUR/GBP	F1	MAE	0.0083	0.0068	0.0062	0.0061	0.0057	0.0052	0.005	0.0048	0.0045	0.0041	0.0036
		MAPE	195.55%	132.76%	130.17%	115.26%	101.27%	95.26%	92.47%	89.16%	86.35%	82.98%	77.55%
		RMSE	0.0098	0.0094	0.0089	0.0087	0.0087	0.0085	0.0082	0.0079	0.0076	0.0075	0.0071
		THEIL-U	0.9104	0.9042	0.8610	0.7854	0.7325	0.6957	0.6688	0.6471	0.6124	0.6009	0.5746
	F2	MAE	0.0092	0.0082	0.0071	0.007	0.0069	0.0066	0.0063	0.0061	0.0058	0.0054	0.0049
		MAPE	212.52%	138.15%	135.22%	118.17%	107.39%	102.62%	99.14%	95.81%	91.29%	87.33%	82.64%
		RMSE	0.0105	0.0101	0.0099	0.0096	0.0095	0.0092	0.0089	0.0088	0.0086	0.0084	0.0081
		THEIL-U	0.9744	0.9055	0.9007	0.8961	0.8777	0.8551	0.8127	0.7806	0.7624	0.7217	0.6749
	F3	MAE	0.008	0.0065	0.006	0.0057	0.0056	0.0051	0.0047	0.0045	0.0042	0.0039	0.0035
		MAPE	191.47%	120.44%	119.69%	109.79%	101.18%	93.88%	90.24%	87.29%	84.17%	79.32%	72.74%
		RMSE	0.0095	0.0091	0.0087	0.0085	0.0082	0.0081	0.0078	0.0077	0.0074	0.0072	0.0068
		THEIL-U	0.8817	0.9015	0.8367	0.7721	0.7129	0.6715	0.6328	0.6110	0.5859	0.5584	0.5226
EUR/CHF	F1	MAE	0.0079	0.0073	0.0072	0.0069	0.0068	0.0067	0.0064	0.0062	0.0058	0.0054	0.0051
		MAPE	175.26%	161.25%	160.25%	135.47%	128.49%	115.28%	108.48 %	102.74%	95.24%	92.36%	87.27%
		RMSE	0.0082	0.0081	0.0076	0.0076	0.0074	0.0071	0.0067	0.0065	0.0064	0.0061	0.0056
		THEIL-U	0.9459	0.9055	0.8091	0.7668	0.7416	0.7301	0.7219	0.7018	0.6654	0.6359	0.6027
	F2	MAE	0.0088	0.0077	0.0076	0.0072	0.0070	0.0068	0.0068	0.0066	0.0063	0.0061	0.0057
		MAPE	195.23%	164.56%	162.28%	152.25%	147.18%	142.29%	131.17%	122.78%	109.57%	99.27%	91.39%
		RMSE	0.0088	0.0084	0.0081	0.0079	0.0078	0.0075	0.0072	0.0069	0.0066	0.0062	0.0059
		THEIL-U	0.9551	0.9074	0.8661	0.8321	0.8189	0.7759	0.7443	0.7155	0.6819	0.6452	0.6158
	F3	MAE	0.0074	0.0069	0.0064	0.0061	0.0059	0.0059	0.0057	0.0056	0.0054	0.0052	0.0048
		MAPE	161.66%	139.27%	138.33%	134.67%	124.25%	110.93%	99.14%	96.78%	94.08%	90.52%	84.91%
		RMSE	0.0079	0.0075	0.0069	0.0068	0.0066	0.0064	0.0061	0.0059	0.0058	0.0056	0.0052
		THEIL-U	0.9229	0.8906	0.7411	0.7298	0.7025	0.6729	0.6414	0.6221	0.6179	0.5815	0.5504

Table 5: Summary of Out-of-Sample Statistical Performances

			RW	ARMA	STAR	MLP	RNN	PSN	AVERAGE	LASSO	KALMAN	GP
EUR/USD	F1	MDM_1	-12.71	-11.24	-10.97	-9.37	-9.13	-8.08	-7.95	-6.25	-5.15	-4.26
		MDM_2	-15.85	-14.37	-13.49	-12.64	-11.97	-10.05	-8.57	-7.06	-6.53	-5.56
	F2	MDM_1	-14.08	-13.19	-12.91	-11.18	-10.27	-8.57	-8.16	-7.59	-6.87	-6.31
		MDM_2	-17.28	-15.21	-14.08	-13.57	-12.37	-10.58	-9.18	-8.17	-9.25	-8.19
	F3	MDM_1	-11.39	-10.23	-9.25	-7.69	-7.81	-6.28	-5.24	-4.38	-4.09	-3.67
		MDM_2	-13.77	-12.19	-10.88	-10.32	-10.11	-9.84	-7.93	-6.81	-5.48	-4.39
EUR/GBP	F1	MDM_1	-13.22	-12.95	-11.57	-11.02	-10.38	-9.51	-9.26	-8.41	-7.32	-6.17
		MDM_2	-14.27	-14.08	-13.47	-13.19	-12.87	-11.28	-10.55	-9.97	-8.75	-7.28
	F2	MDM_1	-14.97	-14.74	-14.09	-13.67	-12.93	-11.96	-10.57	-9.85	-9.07	-8.14
		MDM_2	-16.33	-15.77	-15.13	-14.69	-13.85	-13.42	-12.38	-11.46	-10.75	-9.26
	F3	MDM_1	-12.12	-11.87	-11.28	-10.78	-10.27	-9.48	-8.89	-8.38	-7.17	-5.84
		MDM_2	-14.05	-13.66	-12.67	-12.13	-11.86	-11.19	-10.53	-9.67	-9.23	-8.61
EUR/GBP	F1	MDM_1	-14.54	-14.21	-13.74	-13.27	-12.48	-11.33	-10.26	-9.57	-8.46	-6.98
		MDM_2	-17.02	-16.74	-16.14	-15.22	-14.88	-14.26	-13.56	-12.82	-10.44	-9.41
	F2	MDM_1	-15.29	-14.85	-14.27	-13.85	-13.26	-12.27	-11.55	-10.73	-9.57	-8.52
		MDM_2	-17.61	-16.86	-16.69	-15.37	-14.97	-14.61	-13.78	-13.08	-10.81	-9.96
	F3	MDM_1	-13.84	-13.22	-12.07	-11.85	-10.99	-10.42	-9.87	-9.12	-7.68	-6.37
		MDM_2	-16.28	-15.87	-15.31	-14.27	-13.68	-12.89	-11.54	-10.59	-9.44	-8.12

Note: MDM_1 and MDM_2 are the statistics computed for the MSE and MAE loss function respectively.

Table 6: Summary results of Modified Diebold-Mariano statistics for MSE and MAE loss functions

		TRADITIONAL STRATEGIES			NEURAL NETWORKS			FORECAST COMBINATIONS				
EUR/USD		**RW**	**ARMA**	**STAR**	**MLP**	**RNN**	**PSN**	**AVERAGE**	**LASSO**	**KALMAN**	**GP**	**SVR**
F1	**Information Ratio**	-0.11	0.21	0.32	1.03	1.35	1.58	1.46	1.68	1.93	2.05	2.10
	Sharpe Ratio	-0.16	0.16	0.27	0.97	1.29	1.52	1.41	1.62	1.88	2.00	2.05
	Annualised Return (including costs)	-1.18%	2.29%	3.41%	9.15%	12.08%	14.49%	14.68%	16.23%	18.05%	19.94%	22.18%
	Hit Ratio	46.57%	51.52%	52.19%	54.78%	55.05%	55.26%	55.82%	56.74%	58.29%	59.23%	61.12%
	Maximum Drawdown	-15.57%	-18.39%	-16.55%	-15.18%	-14.73%	-13.25%	-12.37%	-11.79%	-11.81%	-10.91%	-10.82%
F2	**Information Ratio**	-0.37	0.15	0.27	0.80	0.83	1.14	1.32	1.56	1.66	1.68	1.73
	Sharpe Ratio	-0.41	0.11	0.23	0.75	0.77	1.09	1.27	1.51	1.61	1.62	1.68
	Annualised Return (including costs)	-4.52%	1.85%	3.02%	7.81%	8.22%	11.26%	12.08%	14.14%	15.37%	16.17%	18.43%
	Hit Ratio	45.09%	51.68%	52.86%	53.92%	54.05%	55.02%	55.73%	56.36%	56.82%	57.43%	58.92%
	Maximum Drawdown	-21.42%	-12.94%	-18.49%	-13.73%	-14.21%	-12.88%	-12.44%	-12.03%	-11.95%	-12.15%	-11.94%
F3	**Information Ratio**	0.02	0.38	0.51	1.14	1.46	1.68	1.81	2.00	2.01	2.28	2.77
	Sharpe Ratio	-0.02	0.33	0.46	1.09	1.41	1.63	1.76	1.95	1.96	2.23	2.71
	Annualised Return (including costs)	0.27%	3.77%	5.51%	11.26%	14.08%	16.41%	17.32%	19.91%	20.19%	22.62%	25.37%
	Hit Ratio	47.38%	51.88%	53.30%	54.79%	55.12%	55.81%	56.12%	58.28%	58.46%	59.60%	61.47%
	Maximum Drawdown	-19.45%	-11.39%	-13.88%	-11.76%	-11.23%	-11.65%	-10.83%	-10.67%	-10.83%	-10.84%	-10.13%
EUR/GBP		**RW**	**ARMA**	**STAR**	**MLP**	**RNN**	**PSN**	**AVERAGE**	**LASSO**	**KALMAN**	**GP**	**SVR**
F1	**Information Ratio**	-0.18	0.23	0.29	1.07	1.32	1.50	1.42	1.50	1.86	2.03	2.06
	Sharpe Ratio	-0.22	0.17	0.24	1.01	1.26	1.44	1.37	1.45	1.80	1.97	2.01
	Annualised Return (including costs)	-2.36%	2.09%	2.89%	9.07%	11.68%	13.55%	15.18%	15.89%	18.24%	19.35%	20.85%
	Hit Ratio	46.74%	50.95%	51.04%	54.07%	54.95%	55.64%	56.09%	56.80%	58.26%	58.84%	59.42%
	Maximum Drawdown	-14.67%	-13.42%	-13.59%	-14.25%	-14.73%	-14.05%	-13.19%	-13.08%	-11.55%	-10.88%	-10.47%
F2	**Information Ratio**	-0.25	0.17	0.22	0.95	1.16	1.43	1.37	1.41	1.51	1.65	1.89
	Sharpe Ratio	-0.29	0.12	0.18	0.89	1.11	1.37	1.33	1.36	1.46	1.59	1.84
	Annualised Return (including costs)	-3.27%	1.83%	2.45%	8.15%	10.22%	13.08%	14.44%	14.86%	15.02%	15.35%	18.75%
	Hit Ratio	43.22%	45.58%	50.17%	54.19%	54.76%	55.23%	55.88%	55.91%	56.74%	57.41%	58.87%
	Maximum Drawdown	-14.01%	-10.59%	-13.12%	-13.95%	-14.02%	-14.55%	-13.02%	-12.26%	-12.41%	-11.34%	-11.02%
F3	**Information Ratio**	-0.08	0.34	0.39	1.18	1.27	1.54	1.75	1.66	1.93	2.22	2.57
	Sharpe Ratio	-0.12	0.28	0.35	1.12	1.22	1.49	1.69	1.61	1.88	2.17	2.52
	Annualised Return (including costs)	-1.07%	3.02%	4.28%	11.25%	12.18%	14.22%	15.85%	16.47%	19.32%	20.55%	23.27%
	Hit Ratio	47.98%	53.05%	53.86%	54.82%	55.12%	55.92%	56.34%	57.02%	59.05%	59.63%	62.16%
	Maximum Drawdown	-13.38%	-16.73%	-13.44%	-14.79%	-14.62%	-14.18%	-12.27%	-12.17%	-12.97%	-12.24%	-11.67%
EUR/CHF		**RW**	**ARMA**	**STAR**	**MLP**	**RNN**	**PSN**	**AVERAGE**	**LASSO**	**KALMAN**	**GP**	**SVR**
F1	**Information Ratio**	-0.16	0.38	0.28	1.43	1.45	1.52	1.67	1.65	1.85	1.92	2.22
	Sharpe Ratio	-0.20	0.23	0.24	1.38	1.40	1.47	1.62	1.60	1.80	1.87	2.17
	Annualised Return (including costs)	-1.89%	2.56%	3.15%	13.55%	14.02%	15.28%	16.85%	17.07%	17.55%	18.38%	21.82%
	Hit Ratio	48.52%	51.84%	52.02%	55.15%	55.73%	55.94%	56.67%	57.22%	57.68%	58.44%	59.60%
	Maximum Drawdown	-17.22%	-17.05%	-14.99%	-12.56%	-12.47%	-11.98%	-11.05%	-11.15%	-10.49%	-10.42%	-10.21%
F2	**Information Ratio**	-0.21	0.24	0.24	1.07	1.15	1.24	1.39	1.48	1.64	1.85	1.99
	Sharpe Ratio	-0.25	0.18	0.19	1.02	1.10	1.20	1.34	1.44	1.59	1.80	1.94
	Annualised Return (including costs)	-2.26%	2.27%	2.68%	11.65%	12.24%	13.27%	15.19%	15.84%	16.21%	17.67%	19.24%
	Hit Ratio	47.39%	51.73%	51.84%	54.88%	55.09%	55.23%	56.09%	56.98%	57.45%	57.93%	58.48%
	Maximum Drawdown	-13.44%	-11.82%	-12.14%	-13.01%	-13.25%	-13.47%	-12.72%	-12.54%	-11.85%	-11.14%	-10.53%
F3	**Information Ratio**	-0.20	0.28	0.32	1.37	1.40	1.58	1.70	1.81	1.90	2.22	2.57
	Sharpe Ratio	-0.24	0.24	0.28	1.32	1.35	1.53	1.65	1.76	1.85	2.16	2.52
	Annualised Return (including costs)	-2.36%	3.06%	3.84%	14.21%	14.87%	16.12%	17.02%	17.55%	18.05%	20.28%	23.04%
	Hit Ratio	48.71%	51.85%	53.96%	55.12%	55.33%	56.02%	56.95%	57.44%	58.72%	59.40%	62.06%
	Maximum Drawdown	-13.05%	-12.58%	-13.21%	-13.04%	-12.56%	-12.51%	-11.14%	-11.26%	-11.21%	-11.08%	-10.67%

Table 7: Summary of Out-of-Sample Trading Performances for the EUR/USD, EUR/GBP and EUR/CHF for each exercise

	Extremely Low Vol.	Medium Low Vol.	Lower Low Vol.	Lower High Vol.	Medium High Vol.	Extremely High Vol.
L1	2	1.5	1	1	0.5	0

Table 8: Classification of Volatility Leverage (L1) in sub-periods

	Extremely Low δ	Medium Low δ	Lower Low δ	Lower High δ	Medium High δ	Extremely High δ
$L2^{+}$	0	0.5	1	1	1.5	2
$L2^{-}$	2	1.5	1	1	0.5	0

Table 9: Classification of Index Leverage ($L2^{+}$ and $L2^{-}$) in sub-periods

		TRADITIONAL STRATEGIES			NEURAL NETWORKS			FORECAST COMBINATIONS				
EUR/USD		RW	ARMA	STAR	MLP	RNN	PSN	AVERAGE	LASSO	KALMAN	GP	SVR
F1	Information Ratio	-0.06	0.33	0.46	1.32	1.43	1.71	1.68	1.92	2.04	2.30	2.60
	Sharpe Ratio	-0.10	0.28	0.41	1.27	1.38	1.65	1.63	1.87	1.99	2.24	2.55
	Annualised Return (including costs)	-0.58%	3.42%	4.54%	12.33%	13.26%	15.67%	15.78%	18.35%	19.37%	21.58%	25.23%
	Maximum Drawdown	-14.66%	-14.29%	-13.85%	-14.08%	-14.13%	-13.03%	-11.18%	-10.43%	-10.17%	-10.07%	-9.86%
F2	Information Ratio	-0.10	0.28	0.27	1.00	1.11	1.33	1.35	1.71	1.82	2.04	2.42
	Sharpe Ratio	-0.14	0.23	0.34	0.95	1.06	1.27	1.30	1.66	1.76	1.98	2.37
	Annualised Return (including costs)	-1.29%	3.02%	3.96%	9.95%	10.97%	12.87%	13.43%	16.26%	17.15%	18.67%	22.59%
	Maximum Drawdown	-20.56%	-18.15%	-17.88%	-13.27%	-13.57%	-12.41%	-11.89%	-11.57%	-11.04%	-10.85%	-10.21%
F3	Information Ratio	0.04	0.36	0.79	1.38	1.71	1.95	1.92	2.08	2.29	2.47	2.80
	Sharpe Ratio	-0.01	0.34	0.74	1.32	1.66	1.90	1.87	2.03	2.24	2.42	2.74
	Annualised Return (including costs)	0.44%	3.96%	7.84%	12.98%	16.44%	18.25%	18.74%	20.57%	21.94%	23.28%	27.42%
	Maximum Drawdown	-15.54%	-13.24%	-13.15%	-11.12%	-10.92%	-10.97%	-10.41%	-10.08%	-9.76%	-9.57%	-9.48%
EUR/GBP		RW	ARMA	STAR	MLP	RNN	PSN	AVERAGE	LASSO	KALMAN	GP	SVR
F1	Information Ratio	-0.08	0.27	0.35	1.19	1.20	1.38	1.51	1.56	1.86	2.02	2.25
	Sharpe Ratio	-0.11	0.21	0.30	1.14	1.15	1.33	1.47	1.52	1.81	1.93	2.20
	Annualised Return (including costs)	-1.08%	2.95%	4.11%	13.18%	14.05%	15.25%	16.89%	17.24%	19.38%	20.15%	23.15%
	Maximum Drawdown	-13.95%	-14.35%	-12.03%	-11.45%	-11.34%	-11.12%	-10.68%	-10.11%	-9.58%	-9.68%	-10.01%
F2	Information Ratio	-0.12	0.23	0.24	1.09	1.36	1.44	1.57	1.56	1.78	2.06	2.22
	Sharpe Ratio	-0.15	0.19	0.20	1.04	1.31	1.39	1.52	1.52	1.73	2.01	2.17
	Annualised Return (including costs)	-1.76%	2.67%	2.88%	11.36%	13.54%	14.17%	16.01%	16.23%	17.25%	18.67%	21.18%
	Maximum Drawdown	-14.18%	-13.34%	-13.64%	-14.21%	-14.62%	-14.88%	-13.15%	-12.75%	-11.61%	-11.02%	-10.77%
F3	Information Ratio	0.06	0.29	0.34	1.38	1.41	1.55	1.79	1.88	2.02	2.30	2.51
	Sharpe Ratio	0.02	0.24	0.30	1.33	1.37	1.50	1.74	1.83	1.97	2.25	2.46
	Annualised Return (including costs)	0.71%	3.25%	4.41%	14.78%	15.09%	16.26%	18.68%	20.67%	21.55%	23.87%	25.59%
	Maximum Drawdown	-14.56%	-14.44%	-13.45%	-13.78%	-13.58%	-12.39%	-10.02%	-10.25%	-10.35%	-10.15%	-10.11%
EUR/CHF		RW	ARMA	STAR	MLP	RNN	PSN	AVERAGE	LASSO	KALMAN	GP	SVR
F1	Information Ratio	-0.11	0.30	0.34	1.40	1.55	1.67	1.70	1.85	1.78	2.06	2.31
	Sharpe Ratio	-0.15	0.24	0.29	1.36	1.50	1.62	1.65	1.80	1.74	2.01	2.29
	Annualised Return (including costs)	-1.47%	3.27%	3.87%	15.21%	15.95%	16.87%	17.21%	18.81%	19.38%	21.41%	23.48%
	Maximum Drawdown	-16.42%	-13.94%	-12.14%	-11.89%	-11.13%	-11.01%	-11.32%	-11.24%	-10.83%	-10.31%	-10.14%
F2	Information Ratio	-0.18	0.25	0.28	1.22	1.29	1.38	1.51	1.62	1.64	1.82	2.09
	Sharpe Ratio	-0.22	0.21	0.23	1.17	1.25	1.34	1.46	1.57	1.60	1.77	2.04
	Annualised Return (including costs)	-2.05%	2.93%	3.15%	13.21%	13.84%	14.58%	16.03%	16.84%	17.86%	18.25%	21.15%
	Maximum Drawdown	-15.14%	-13.77%	-12.67%	-11.56%	-11.89%	-11.23%	-10.72%	-10.41%	-10.11%	-9.98%	-9.62%
F3	Information Ratio	-0.09	0.32	0.39	1.52	1.58	1.68	1.81	1.89	2.19	2.41	2.58
	Sharpe Ratio	-0.14	0.25	0.34	1.47	1.53	1.64	1.77	1.84	2.14	2.36	2.53
	Annualised Return (including costs)	-1.14%	3.48%	3.96%	16.08%	16.51%	17.54%	19.54%	20.15%	22.19%	24.51%	26.17%
	Maximum Drawdown	-12.45%	-13.12%	-12.73%	-11.67%	-11.69%	-11.56%	-10.32%	-10.29%	-9.87%	-9.65%	-9.48%

Note: Not taken into account the interest that could be earned during times where the capital is not traded (non-trading days) or not fully invested and could therefore be invested.

Table 10: Summary of Out-of-Sample Trading Performances for the EUR/USD, EUR/GBP and EUR/CHF after leverage - final results

EUR/USD			EUR/GBP			EUR/CHF		
0.0001***			0.0001***			0***		
F1			**F2**			**F3**		
EUR/USD	**EUR/GBP**	**EUR/CHF**	**EUR/USD**	**EUR/GBP**	**EUR/CHF**	**EUR/USD**	**EUR/GBP**	**EUR/CHF**
0.0001***	0.0001***	0.0001***	0.0001***	0***	0.0001***	0.0001***	0***	0.0001***

Note: The top three values of the table are the p-values of the ADF tests for the whole sample (01/02/1999 - 31/12/2012).

Table A.1: The p-values of the ADF tests for each exchange rate

	Parameters	MLP			RNN			PSN		
	Exercise	F1	F2	F3	F1	F2	F3	F1	F2	F3
EUR/USD	Learning algorithm	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent
	Learning rate	0.001	0.002	0.008	0.002	0.003	0.005	0.003	0.003	0.007
	Momentum	0.003	0.004	0.009	0.004	0.004	0.007	0.005	0.005	0.009
	Iteration steps	60000	80000	80000	75000	80000	90000	80000	65000	85000
	Initialisation of weights	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)
	Input nodes	8	9	7	9	9	7	8	8	7
	Hidden nodes	6	7	6	5	6	6	4	6	6
	Output node	1	1	1	1	1	1	1	1	1
EUR/GBP	Learning algorithm	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent
	Learning rate	0.002	0.002	0.005	0.002	0.003	0.004	0.003	0.002	0.005
	Momentum	0.003	0.003	0.006	0.004	0.005	0.005	0.005	0.004	0.006
	Iteration steps	50000	70000	80000	60000	65000	80000	40000	35000	75000
	Initialisation of weights	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)
	Input nodes	9	9	6	9	7	7	7	7	9
	Hidden nodes	5	7	5	6	6	5	5	4	3
	Output node	1	1	1	1	1	1	1	1	1
EUR/CHF	Learning algorithm	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Gradient descent
	Learning rate	0.001	0.001	0.005	0.002	0.001	0.004	0.002	0.002	0.001
	Momentum	0.004	0.004	0.006	0.002	0.003	0.005	0.005	0.006	0.003
	Iteration steps	70000	50000	60000	50000	35000	40000	55000	70000	70000
	Initialisation of weights	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)	N(0,1)
	Input nodes	7	6	6	7	8	7	7	8	8
	Hidden nodes	6	4	3	6	7	3	5	7	5
	Output node	1	1	1	1	1	1	1	1	1

Table B.1: The NNs training characteristics

	GENETIC PROGRAMMING PARAMETERS		
	EUR/USD	**EUR/GBP**	**EUR/CHF**
Population Size	200	200	200
Termination Criterion	100000	100000	100000
Max. tree depth	6	6	6
Function Set	+, -, *, /, ^, ^2, ^3, ^1/2, ^1/3, Exp, If, sin, cos, tan	+, -, *, /, ^, ^2, ^3, ^1/2, ^1/3, Exp, If, sin, cos, tan	+, -, *, /, ^, ^2, ^3, ^1/2, ^1/3, Exp, If, sin, cos, tan
Tournament Size	**4**	**4**	**4**
Crossover trials	1	1	1
Mutation Probability	0.75	0.75	0.75

Table C.1: GP parameters' setting for each exchange rate.

STATISTICAL PERFOMANCE MEASURES	DESCRIPTION
Mean Absolute Error	$MAE = (\frac{1}{n})\sum_{\tau=t+1}^{t+n}\left\|\hat{Y}_\tau - Y_\tau\right\|$ *with* Y_τ *being the actual value and* $\hat{Y}_\tau$ *the forecasted value*
Mean Absolute Percentage Error	$MAPE = \frac{1}{n}\sum_{\tau=t+1}^{t+n}\left\|\frac{Y_\tau - \hat{Y}_\tau}{Y_\tau}\right\|$
Root Mean Squared Error	$RMSE = \sqrt{\frac{1}{n}\sum_{\tau=t+1}^{t+n}(\hat{Y}_\tau - Y_\tau)^2}$
Theil-U	$Theil-U = \frac{\sqrt{(\frac{1}{n}\sum_{\tau=t+1}^{t+n}(\hat{Y}_\tau - Y_\tau)^2}}{\sqrt{\frac{1}{n}\sum_{\tau=t+1}^{t+n}\hat{Y}_\tau^2} + \sqrt{\frac{1}{n}\sum_{\tau=t+1}^{t+n}Y_\tau^2}}$

Table D.1: Statistical Performance Measures and Calculation

TRADING PERFOMANCE MEASURES	DESCRIPTION
Annualised Return	$R^A = 252 * \frac{1}{N} * (\sum_{t=1}^{N} R_t)$ where R_t the daily returns
Hit Ratio	$H = \frac{WinningTrades}{TotalTrades}$
Information Ratio	$IR = \frac{R^A}{\sigma^A}$
Sharpe Ratio	$SR = \frac{R^A - r_f^A}{\sigma^A}$ where r_f^A is the risk free rate p.a.*
Maximum Drawdown	Maximum negative value of $\sum(R_t)$ over the period $MD = \underset{i=1,\cdots,t;t=1,\cdots,N}{Min}\left(\sum_{j=i}^{t} R_j\right)$

**Note: In this case the risk free rate is 0.504% p.a., which corresponds to the Euribor rate at the time of the calculation.*

Table D.2: Trading Performance Measures and Calculation